**UNIVERSITE SAAD DAHLAB DE BLIDA1**

**Faculté des Sciences**
Département d'Informatique

# MEMOIRE DE MAGISTER

Spécialité : Informatique Répartie et Mobile (IRM)

## METHODES D'ARBRES EN INDEXATION ET RECHERCHE D'IMAGES PAR SIMILITUDE VISUELLE

Par

**KHALFI Ali**

Présentée devant le jury composé de :

| | | |
|---|---|---|
| S.OUKID | Maître de Conférences (A), université de Blida1 | Présidente |
| L.HAMAMI | Professeur, école national polytechnique, Alger | Examinatrice |
| W.K.HIDOUCI | Professeur, école supérieure d'informatique, Alger | Examinateur |
| N. BENBLIDIA | Maître de Conférences (A), université de Blida1 | Promotrice |
| A.CHERIF-ZAHAR | Maître Assistant (A), université de Blida1 | Invité |

Blida, mars 2015

# RESUME

Actuellement les systèmes d'indexation et de recherche d'images ou de vidéos se sont basés sur la description et l'indexation du contenu visuel. En réalité une vidéo est une séquence d'image. Ces images peuvent être regroupées entre elles selon leurs degrés de similarité en des unités élémentaires appelées ''plan vidéo''. La procédure de regroupement des images similaires d'une vidéo est connue souvent sous le nom '' segmentation temporelle en plans de vidéo''.

Le but de notre travail consiste à proposer une nouvelle technique de segmentation en plans de vidéo dans le contexte de la recherche d'image par similitude visuelle. Cette technique est basée sur un algorithme qui repose sur l'exploitation de la méthode d'arbre quaternaire comme structure de données pour l'indexation de l'image. La segmentation en plans de vidéo exploite une combinaison de plusieurs descripteurs issus de l'histogramme scalable de couleur (SCD) et de l'histogramme d'orientation des contours (EHD) de la norme MPEG 7([1]). Les descripteurs qui sont sauvegardés à différents niveaux de l'arbre quaternaire permettent de réaliser l'indexation de l'image. Une distance est définie pour comparer deux images représentées respectivement par deux arbres quaternaires. Les résultats expérimentaux ont permis de montrer la robustesse offerte par notre algorithme.

**Mots clés :** Segmentation temporelle, Arbre Quaternaire, Changement de Plan Vidéo, MPEG-7 , Descripteurs Visuels.

([1]) Moving Picture Experts Group, groupe de travail de l'ISO/IEC chargé du développement de standards d'encodage audio et vidéo.

# ABSTRACT

Currently systems indexing and searching images or videos are based on the description and indexing of video content. In reality a video is image sequence. These images can be grouped together according to their degrees of similarity in basic units called ' video shot'. The clustering procedure similar images of a video is often known as 'temporal segmentation video shot'.

The goal of our work is to propose a new technique for video shot segmentation in the context of image search by visual similarity. This technique is based on an algorithm which is based on the operation of the quad-tree data structure as a method for indexing the image. The video shot segmentation uses a combination of descriptors from the scalable color histogram (SCD) and the orientation histogram contours (EHD) of the MPEG 7 standard . The descriptors are stored at different levels of the quad-tree data structure for effecting the indexing of the image. A distance is defined to compare two images represented respectively by two quaternary trees. Experimental results have shown the robustness offered by our algorithm.



**Keywords**

Temporal segmentation, quad-tree, changing video shots, MPEG -7, visual descriptors.

# مـلـخــص

ان انظمة الفهرسة و البحث عن الصور و مقاطع الفيديو تعتمد حاليا على وصف و فهرسة على ما تحتويه هذه الاخيرة ,فكما نعلم ان شريط الفيديو يتكون من عدة صور متسلسلة فيما بينها ملتقطة عبر مراحل زمنية معينة,فيمكننا تجميع هذه الصور الجد متشابهة من حيث محتواها الى مجموعات تتمثل في وحدات اساسية تدعى بلقطات فيديو ضرورية في تكوين شريط الفيديو.فعملية تشكيل هذه الوحدات تسمى بالتجزئة الزمانية للفيديو الى لقطات .

و الهدف من هذا البحث هو اقتراح تقنية جديدة مطبقة على ملفات الفيديو من اجل الحصول على لقطات زمنية متفرقة في سياق البحث عن الصور من خلال التشابه البصري بين الصور المكونة للفيديو ومن حيث ما تحتويه هذه الصور من الوان و نسيج لوني .تستند هذه التقنية على خوارزمية التي بدورها تعتمد على الشجرة الرباعية كبنية لتخزين بيانات الصور وفهرستها . كما تستخدم تقنية التجزئة الزمنية اللفيديو الى لقطات المقترحة الجمع بين الواصفات ,التي تتمثل في واصف اللون البياني(SCD),و كفاف التوجه اللون البياني(EHD) المعتمدين من طرف MPEG 7 و واصف كشف التوجه , حيث يتم تخزين الواصفات المستخرجة من الصور على مستويات مختلفة من الشجرة لإحداث فهرسة الصور .تستخدم هذه التقنية واصفات من اللون البياني وكفاف التوجه اللون البياني و واصف كشف التوجه. ثم يتم تعريف المسافة للمقارنة بين الصور الممثلة على التوالي من قبل اثنين من الأشجار الرباعية. ولقد أظهرت النتائج التجريبية مدى متانة , قوة و نجاح هذه التقنية التي تقدم ذكرها.

**كلمات مفتاحية :**

*تجزئة زمنية , شجرة رباعية , تغير لقطات للفيديو* ,MPEG -7, *الواصفات البصرية*.

**DEDICACES**

*A mes parents …*

# REMERCIEMENTS

En particulier, Je tiens à remercier Melle N.Benblidia maître de conférences à l'USDB qui m'a honorée de sa confiance en acceptant d'être ma promotrice au cours de la réalisation de mon mémoire de magister. Elle m'a permis de poursuive mon travail de recherche dans un esprit scientifique rigoureux. Son écoute attentive et ininterrompue, ainsi que ses vastes connaissances dans le présent domaine m'ont été d'une aide précieuse.

Je souhaite remercier profondément Mme S.Oukid maître de conférences à l'USDB qui m'a fait l'honneur de présider mon jury de soutenance. J'exprime mes vif remerciement à Mr W.K.Hidouci professeur à l'école supérieure d'informatique et à Mme L.Hamami professeur à l'école nationale polytechnique alger, d'avoir accepté de juger mon travail.

Ma gratitude a Mr A.cherif-Zahar maître assistant à l'USDB pour son soutient et ses conseils qui m'ont aidé à la réalisation de ce mémoire.

Je remercie également Mr C.Adidou, mes collègues, mes amis pour leurs encouragements, leurs aides et leur support.

# LISTE DES FIGURES

## LISTE DES TABLEAUX

**TABLE DES MATIÈRES**

# INTRODUCTION GENERALE

Le développement rapide de la technologie permet au grand public d'utiliser des équipements multimédias sophistiqués. La quantité d'information audiovisuelle accessible croit ainsi de façon spectaculaire avec l'apparition de l'internet à haut débit et de la télévision numérique. L'utilisateur a besoin de rechercher des fichiers audiovisuels dans une banque de documents, de plus en plus importante. Cela nécessite alors le développement de techniques efficaces pour la recherche d'information multimédia et en particulier la recherche d'images ; ce qui induit une phase d'indexation du document audiovisuel.

Les techniques d'indexation des vidéos s'appuyaient initialement sur une approche textuelle et manuelle : les images étaient annotées par des mots clefs. Cependant la recherche des vidéos reste limitée à ces mots clefs et ne permet pas d'avoir des résultats satisfaisants. Ceci est dû notamment à l'augmentation rapide du volume des données, ce qui rend l'annotation manuelle de moins en moins envisageable, mais aussi au contenu subjectif intrinsèque des images ce qui rend l'indexation dépendante de son auteur. Ces limites ont donc donné naissance à un nouveau type d'indexation basée sur l'information portée par les images elles-mêmes et non plus seulement sur des mots clefs associés à ces dernières : c'est l'indexation par le contenu [1].

L'indexation et la recherche des vidéos par le contenu visuel visent à extraire directement de la vidéo l'information nécessaire à sa caractérisation. Une séquence vidéo est composée d'une ou plusieurs scènes, qui sont composées d'un ou plusieurs plans, chacun contient une ou plusieurs images et chaque image contient un ou plusieurs objets.

Les méthodes d'indexation des séquences vidéo basées sur le contenu visuel utilisent l'analyse des images qui les composent. La première étape principale à suivre est la segmentation élémentaire d'une vidéo qui permet de regrouper les images successives en des unités qui correspondent à des prises de vues de la caméra souvent connues sous le nom de ‘‘plans vidéo’’. Notons que toutes les méthodes de traitement du contenu de la vidéo (la segmentation en scènes, suivi d'objets dans les plans, similitude entre les plans, etc.) se basent sur ces unités et donc la fiabilité de leurs résultats dépend de la précision de détection des plans [2]. De ce fait, les chercheurs se sont intéressés à cette problématique et ont proposé plusieurs techniques pour la segmentation temporelle de vidéos qui est une étape cruciale dans des applications telles que la gestion de bases de données multimédias. Dans ce contexte, nous nous intéressons particulièrement à la détection des changements de plans d'une vidéo.

Dans ce cadre, nous proposons une nouvelle technique de segmentation temporelle en plans vidéo basée sur la méthode d'arbre quaternaire et les descripteurs visuels. La présente structure permet d'indexer spatialement chaque image obtenue après le découpage de la vidéo à traiter. Nous exploitons les descripteurs visuels de la norme MPEG7 pour la caractérisation du contenu visuel des images dans le but de faire la recherche d'image par le contenu et la classification des segments vidéo. Nous avons opté pour l'utilisation de l'histogramme scalable de couleur (SCD) ainsi que l'histogramme d'orientation des contours recommandé par MPEG 7 permettant de différencier les textures non uniformes. Pour caractériser la forme, nous avons choisi la détection des contours de l'image par l'approche de convolution en utilisant le noyau de Sobel. Le cœur de notre travail est décrit par un algorithme qui sera développé au cours de ce mémoire dans le but de détecter les changements de plan d'une vidéo en exploitant la structure d'arbre quaternaire et les descripteurs visuels.

Le manuscrit de notre travail est divisé en quatre chapitres ; nous commençons par un état de l'art où nous aborderons en premier lieu, le concept de la recherche d'image par le contenu avec les différents attributs qui caractérisent l'information portée par l'image. Les descripteurs visuels de la norme MPEG7 seront exposés ainsi que les distances de similarités entre les descripteurs visuels. En

second lieu, nous présenterons les principales étapes de l'indexation et de la recherche des vidéos par le contenu. En se basant sur l'objectif de notre travail, nous allons décrire un panorama des travaux relatifs aux méthodes de la segmentation en plans de la vidéo existante dans la littérature.

Le deuxième chapitre expose les différentes structures d'arbres en indexation et recherche d'image par similitude visuelle. Une étude particulière sera consacrée à la méthode d'arbre quaternaire et l'arbre R ; en effet, ces structures arborescentes sont très utilisées pour la gestion des images similaires ce qui nous facilite l'utilisation de leurs concepts dans notre thème.

Le troisième chapitre présente notre technique pour la segmentation en plans de la vidéo basée sur la structure de l'arbre quaternaire et les descripteurs visuels.

Enfin, nous terminerons par le quatrième chapitre dans le but est de présenter la partie expérimentale et de discuter les différents résultats obtenus par la technique proposée. Une conclusion générale finalise notre travail, résume la contribution du travail effectué, discute les principaux résultats obtenus et fixe les perspectives envisageables.

# CHAPITRE 1
# ÉTAT DE L'ART

## 1.1 Introduction

La principale fonction d'un système d'indexation et recherche d'image était d'indexer et rechercher une image en se basant sur des mots clés, c'est l'approche la plus ancienne. Le système informatique correspondant assigne automatiquement une légende ou des mots clés à une image numérique dans le but d'organiser et de retrouver des images d'intérêts dans une base de données, c'est l'approche la plus ancienne. Les avantages de l'annotation automatique d'images sont que les requêtes peuvent être spécifiées plus naturellement par l'utilisateur, sous forme de requête textuelle. Cependant son application génère plusieurs inconvénients tels que l'ambiguïté inhérente, dépendance du contexte subjectivité de l'indexation, dépendant de la langue et coût d'annotation manuelle.

Dans ce chapitre, nous allons aborder le domaine de l'indexation et la recherche par le contenu. Le présent chapitre est divisé en deux sections. Dans la première section, nous allons donner les principaux concepts de base relatifs à la recherche d'image basée sur le contenu. Nous présentons d'abord les principales phases d'un système de recherche d'image par le contenu ensuite les caractéristiques de base pour la description de l'apparence visuelle des images pour une recherche efficace par le contenu. Puis, nous introduisons les différentes approches de mesure de similarité proposées dans la littérature.

Vu que toutes les méthodes de traitement du contenu de la vidéo se basent sur la recherche de plans vidéos qui est une étape primordiale dans un système d'indexation et recherche de vidéos, la deuxième section de ce chapitre sera consacrée pour présenter un panorama des méthodes de détection de changement de plans dans le domaine de l'indexation et recherche de vidéos. Ce domaine faisant

l'objet de nombreux travaux, nous présenterons le concept de segmentation de la vidéo, segmentation en scène, résumé vidéo et les techniques de segmentation en plans de vidéo, enfin nous terminerons par une conclusion.

## 1.2 Indexation et recherche d'image

### 1.2.1 Les phases d'indexation et recherche d'image par le contenu

L'approche actuelle d'indexation consiste à décrire les images par leurs contenus à l'aide de descripteurs constitués de paramètres de bas niveau relatifs à la couleur, la forme et la texture. Ainsi, à chaque image on fait correspondre un ou des vecteurs caractéristiques formant les index de cette image. La recherche de similarité n'est donc, pas mesurée sur les images directement, mais plutôt sur la base des vecteurs caractéristiques. Étant donnée une requête, un système de recherche d'images doit être capable, sur la base des index dont il dispose, de retrouver les images les plus similaires à une image requête en terme d'une distance donnée. Le but de recherche d'images se ramène donc à rechercher les vecteurs les plus proches voisins à un vecteur requête au sens de similarité donnée dans un espace de vecteurs multidimensionnels. Ainsi, les systèmes de recherche d'images par le contenu se décrivent par la figure 1.1:

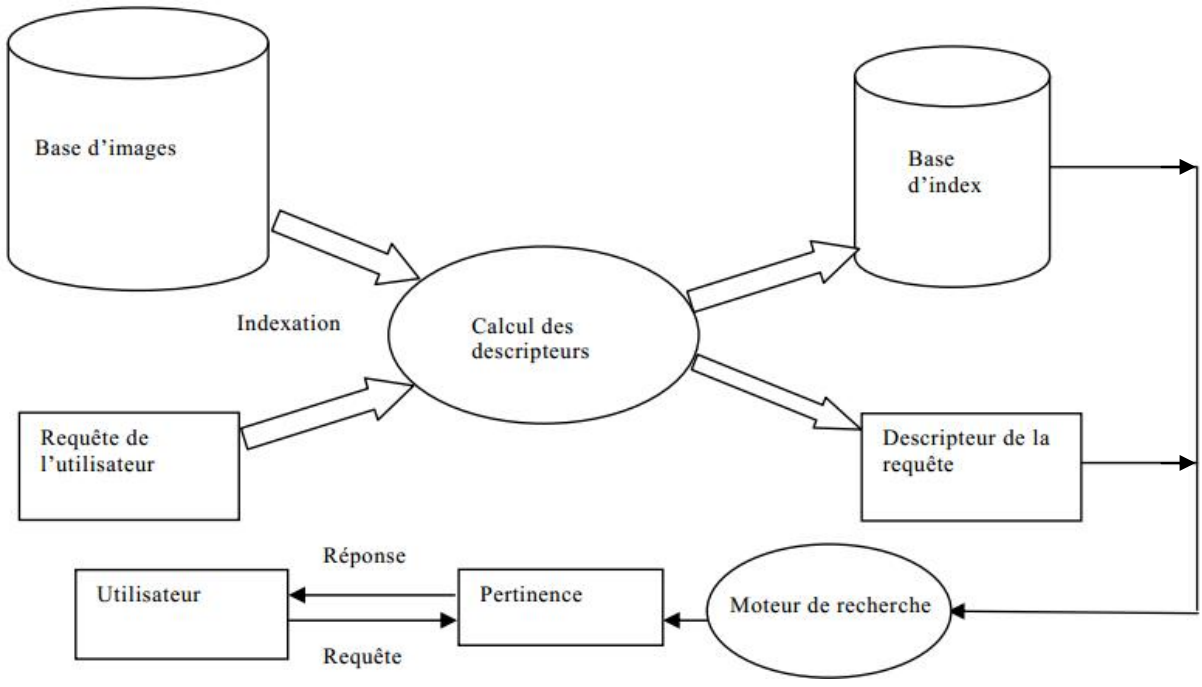


Figure 1.1 : Schéma d'un système de recherche d'image par le contenu [3]

Le système de recherche et l'indexation d'images par le contenu (en anglais : *Content Based Image Retrieval* ou CBIR) s'exécutent en deux étapes : l'étape d'indexation et l'étape de recherche. Dans l'étape d'indexation (phase hors-ligne), des caractéristiques sont automatiquement extraites à partir de l'image et stockées dans un vecteur numérique appelé descripteur visuel. Grâce aux techniques de la base de données, on peut stocker ces caractéristiques et les récupérer rapidement et efficacement. Dans l'étape de recherche (phase en-ligne), le système prend une requête à l'utilisateur et lui donne le résultat correspondant à une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance.

### 1.2.2 L'indexation

Dans un système de recherche d'image par le contenu, le mot indexation comporte deux étapes :

le premier consiste à extraire les caractéristiques visuelles d'image telle que la couleur, la texture ou la forme; ensuite de les représenter par une signature numérique qui s'appelle en général un descripteur visuel sous forme d'un vecteur de dimension N selon les techniques utilisées dans la littérature.

La deuxième étape consiste à indexer la base des vecteurs dans le but d'avoir un temps de réponse minimal pour avoir les images similaires pour une image requête. Dans cette phase, il existe plusieurs méthodes pour l'indexation de la base des vecteurs. Dans la plupart des cas une structure hiérarchique est utilisée pour organiser et faciliter l'accès aux signatures pertinentes tel que les structures arborescentes arbre-R[4].

### 1.2.3 La recherche

Dans cette étape l'utilisateur émet une ou plusieurs requêtes et le système analyse cette dernière pour renvoyer à l'utilisateur le résultat correspondant en une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance.

Il existe deux types de recherche en recherche d'image par le contenu, le premier est la recherche globale qui consiste à comparer les deux descripteurs globaux de deux

images en entier. Et le deuxième type est la recherche locale qui permet decomparer deux images localement selon leurs descripteurs visuels locaux pour chaque région.

### 1.2.4 Attributs et descripteurs d'images

Les images sont des objets numériques très riches en termes d'information. Leur manipulation directe ne permet pas d'obtenir des temps de réponse réalistes pour un système de recherche d'images. Il convient donc d'utiliser une représentation dimensionnelle réduite pour caractériser une image. Ainsi, on va extraire des attributs caractéristiques de l'image à l'aide de fonctions mathématiques et on va les regrouper sous la forme d'un représentant de l'image: le vecteur descripteur de l'image.

#### 1.2.4.1 L'importance des attributs

Les attributs vont représenter l'image, leur choix est déterminant pour la suite de la méthode [5]. Si les attributs sont mal choisis, la méthode de classification donnera de mauvais résultats. Comment choisir de bons attributs ?

Il n'y a pas de réponse générale à cette question, car le choix des attributs va dépendre de ce que l'on souhaite classer. Les attributs sont en général choisis par un expert du domaine des images de la base. L'expert justifie le choix des attributs par son expérience, sur les caractéristiques qui lui semblent importantes et sur le champ applicatif de la méthode de recherche. Suivant l'application développée, l'expert pourra choisir différents attributs. Le choix des attributs est fortement dépendant des images de la base. Ainsi, les attributs qui donnent d'excellents résultats sur une base d'images peuvent donner des résultats médiocres sur une autre base. Il n'y a pas d'attributs universels donnant de bons résultats sur n'importe quelle base d'images. Il existe deux familles d'attributs, les attributs globaux qui sont calculés à partir de l'image entière et les attributs locaux qui sont calculés sur une région de l'image considérée.

Les principaux attributs pour représenter le contenu de l'image sont classés en trois familles : la couleur, la texture et la forme.

### 1.2.4.2 La couleur

Le premier attribut utilisé pour la recherche d'images par le contenu est la couleur. Il est en théorie invariant aux translations et rotations, et change seulement légèrement en cas de changements de la prise de vue ou de l'échelle.
Il existe de nombreuses possibilités d'attributs pour caractériser la couleur : l'histogramme, les moments couleur…avec la même méthode, si on change l'espace de couleur, il peut donner des informations différentes de l'image.

#### 1.2.4.2.1 L'histogramme de couleur

On définit l'histogramme des niveaux de gris d'une image comme étant la fonction $h :[ 0..L-1] \rightarrow N$ qui associe à chaque niveau de gris entre 0 et L-1 la quantité de pixels de l'image qui possèdent cette intensité lumineuse [6].
L'histogramme d'une image peut être représenté par un vecteur dont chaque Composante est un nombre de pixels de niveau de gris correspondant à son indice. Les histogrammes sont faciles et rapides à calculer, et robustes à la rotation et à la translation. Cependant il y a quatre problèmes en utilisant des histogrammes pour l'indexation et la recherche d'images [7]:

a- les histogrammes sont de grandes tailles, donc par conséquent il est difficile de créer une indexation rapide et efficace.
b- les histogrammes ne possèdent pas d'informations spatiales sur les positions des couleurs. Dans certains cas, il y a des images différentes, mais ces images ont les mêmes histogrammes.
c- les histogrammes sont sensibles à de petits changements de luminosité. c'est-à-dire, c'est difficile pour comparer des images similaires dans des conditions différentes.
d- on ne peut pas faire la comparaison partielle des images (objet particulier dans une image), puisqu'on doit calculer globalement l'histogramme sur toute l'image.

### 1.2.4.2.2 Les espaces de couleur

Le mode colorimétrique de fondement pour les applications numériques est basé sur le fonctionnement des écrans vidéo (ordinateur, télévision…) qui restituent les teintes par l'addition en lumière (faisceaux d'électrons) de trois couleurs primaires : le rouge (R), le vert (G) et le bleu (B). Avec ce sous-système une teinte Θ est obtenue par une combinaison linéaire donnée par:

$$\Theta = \alpha * R + \beta * G + \gamma * B \text{ avec } \alpha, \beta, \gamma \in [0,1]^3. \qquad (1.1)$$

Dans le cas d'une image RGB codée sur 8 bits (cas le plus courant), chaque couleur primaire est codée sur un octet et peut donc prendre n'importe quelle valeur dans l'intervalle [0, 255]. Le noir et le blanc correspondent alors respectivement au triplet (0, 0, 0) et (255, 255, 255).

L'espace TSV (Teinte Saturation Valeur - en anglais HSV) est le plus utile pour la segmentation et la reconnaissance et il a été prouvé un espace qui possède des propriétés très intéressantes pour extraire des attributs représentatifs dans le système de recherche des images [5]. Dans cet espace, on peut séparer pour un pixel l'intensité du pixel (valeur) et la couleur du pixel (teinte et saturation).

En plus des espaces RVB, TSV d'autres espaces tels que ceux correspondants à CIE.LUV [8] et HVC [9] ont fait l'objet de nombreuses études.

### 1.2.4.2.3 Les Moments statistiques

Les moments statistiques correspondent aux dominantes caractéristiques de couleur tel que l'espérance, la variance et d'autres moments. On peut calculer l'espérance, la variance, les moments sur chaque composante par les formules suivantes :

$$E = \frac{1}{N}\sum_{j=1}^{N} p_j \qquad (1.2)$$

$$\delta = \left(\frac{1}{N}\sum_{j=1}^{N}\left(p_j - E\right)^2\right)^{\frac{1}{2}} \qquad (1.3)$$

$$s = \left(\frac{1}{N}\sum_{j=1}^{N}\left(p_j - E\right)^3\right)^{\frac{1}{3}} \qquad (1.4)$$

Où E est l'espérance, $\delta$ est la variance, s est le troisième moment.

### 1.2.4.3 La texture

La texture est le second attribut visuel largement utilisé dans la recherche d'images par le contenu après l'attribut couleur, c'est une caractéristique fondamentale des images, en général il n'existe pas une définition précise de la texture ; en vision par ordinateur on parle fréquemment de répétition de motifs similaires. Elle peut être caractérisée par l'attribut de contraste, de directionalité, de régularité et de la périodicité du motif. La texture permet de différencier des régions de couleurs similaires, mais de sémantiques différentes [10].
Plusieurs méthodes sont utilisées pour analyser la texture. On peut citer :

#### 1.2.4.3.1 La matrice de cooccurrences

Cette méthode permet de déterminer la fréquence d'apparition d'un motif formé de deux pixels séparés par une certaine distance *d* dans une direction particulière par rapport à l'horizontale [11], plusieurs statistiques peuvent alors être calculées à partir de la matrice cooccurrences. Nous pouvons citer à titre d'exemple : l'énergie, l'inertie, l'entropie et le contraste donnés respectivement par les équations suivantes :

$$Energie = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(P_{ij}(d,\theta)^2\right) \quad (1.5)$$

$$Inertie = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left((i-j)^2 P_{ij}(d,\theta)\right) \quad (1.6)$$

$$Entropie = -\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(log P_{ij}(d,\theta) P_{ij}(d,\theta)\right) \quad (1.7)$$

$$Contraste = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left((i-j)^2 P_{ij}(d,\theta)\right) \quad (1.8)$$

#### 1.2.4.3.2 La transformée de Fourier

La transformée de Fourier est un outil mathématique qui permet de projeter un signal sur une base de fonctions sinus et cosinus de différentes fréquences [5]. Le signal transformé est la représentation fréquentielle du signal de départ. Il existe plusieurs méthodes d'extraction d'attributs de texture qui sont basées sur la transformée de Fourier.

#### 1.2.4.3.3 Les ondelettes

La transformée en ondelettes est un outil mathématique récent qui décompose un signal en fréquences en conservant une localisation spatiale [12] [13]. Le signal de départ est projeté sur un ensemble de fonctions de base qui varient en fréquences et en espace. Ces fonctions de base s'adaptent aux fréquences du signal à analyser. Cette transformation permet d'avoir une localisation en temps et en fréquences du signal analysé.

En outre, il est à la base de nombreuses analyses de texture, telles que les filtres de Haar. La description de texture à base d'ondelettes est utilisée dans la recherche d'images [5].

### 1.2.4.4. La forme

L'attribut forme est l'un des attributs bas niveau également le plus utilisé pour décrire le contenu visuel des images. Ces attributs sont utilisés pour décrire la structure géométrique générique du contenu visuel. Zhang et al [14] ont proposé de classifier les descripteurs de forme en deux familles :

- ✓ Descripteurs orientés région : ils décrivent les objets selon la distribution spatiale des pixels qui les constituent.
- ✓ Descripteurs orientés contour : ils décrivent les objets selon leur contour externe.

### 1.2.5 Le standard MPEG-7

MPEG-7 est un standard de description de contenu multimédia développé par Moving Picture Experts Groups, groupe de travail de l'ISO/IEC chargé du développement de standards d'encodage audio et vidéo [15]. Ce groupe de travail a déjà créé différents standards d'encodage bien connus et largement utilisés, tels que MPEG-1, MPEG-2, MPEG-3, MPEG-4, qui standardisent l'encodage des documents audiovisuels rendant respectivement possible l'apparition de la vidéo interactive sur le CDROM (VCD, SVCD, DVD), la télévision digitale (satellite, câble ou encore ADSL), le téléchargement, le streaming sur Internet et le multimédia sur mobile. La norme MPEG-7, officiellement nommée -Multimedia Interface Description , fournit un riche ensemble d'outils normalisés pour décrire le contenu multimédia [16].

#### 1.2.5.1 Descripteurs visuels MPEG-7

Les descripteurs visuels de MPEG-7 se réfèrent aux attributs les plus largement utilisés dans le domaine de l'indexation par le contenu. Ces descripteurs visuels permettent d'encoder des informations de couleur, texture, forme et même de mouvement. Le tableau 1.1 résume les différents descripteurs, en fonction de leurs trois types (texture, couleur, forme) [17].

| Couleur | Texture | Forme | Mouvement |
|---|---|---|---|
| Espace de couleur | Histogramme des orientations des contours | Forme - Région | Mouvement de la caméra |
| Quantification de couleur | Texture homogène | Forme - Contour | Trajectoire |
| Histogramme de couleur scalable | Parcours rapide à partir de la texture | Forme – 3D | Mouvement paramétrique 2D |
| Couleur dominante | | | Activité de mouvement |
| Couleur d'un groupe de trames | | | |
| Couleur structurée | | | |
| Distribution spatiale de couleur | | | |

Tableau 1.1 : Les descripteurs visuels MPEG-7 [17].

### 1.2.5.2 Domaines d'application de MPEG-7

MPEG-7 propose de couvrir une gamme d'applications aussi large que possible. Les divers domaines d'application ciblés par MPEG-7 sont [17] :

- ✓ l'archivage radio,
- ✓ la TV et le cinéma,
- ✓ les systèmes d'informations géographiques ou touristiques,
- ✓ le journalisme,
- ✓ l'architecture,
- ✓ le domaine biomédical,
- ✓ le commerce électronique.

Enfin, MPEG-7 se propose également d'aborder un certain nombre d'applications ayant un profil hautement spécialisé, voire professionnel. Dans ce cadre, nous retrouvons des applications telles que :

- téléshopping,
- imagerie satellitaire (Remote Sensing Applications),
- applications éducatives (Educational applications),
- télé-surveillance (Surveillance applications).

## 1.2.6 Vecteur descripteur et mesure de similarité

Le vecteur descripteur est un vecteur caractéristique de l'image construite à partir des attributs extraits de l'image. Il se présente généralement sous forme d'un vecteur à *N* composantes réelles décrivant le contenu visuel d'une image et pouvant être de très grande dimension.

La recherche d'images similaires par le contenu est basée sur la similarité des caractéristiques visuelles telles que la couleur, la texture ou la forme. La fonction distance utilisée pour évaluer la similarité dépend des critères de la recherche, mais également de la représentation des caractéristiques. L'idée principale est généralement d'associer à chaque image un vecteur multidimensionnel représentant les caractéristiques de l'image, et de mesurer la similarité des images en utilisant une fonction de distance entre les vecteurs. Une fonction de distance d(x ,y) entre les points x, y est une métrique si :

- d(x,y) >= 0 (1.9)
- d(x,y) = 0 ⇔ x=y (1.10)
- d(x,y) = d(y,x) (1.11)
- d(x,y) <= d(x,z)+d(z,y) (1.12)

### 1.2.6.1 Distance de similarité

Lorsque les données sont assimilées à des vecteurs, ce qui est souvent le cas, la distance de Minkowski est fréquemment employée. Elle est donnée par [18] :

$$d_m(V_1, V_2) = \left( \sum_{i=1}^{N} W_i \; X \; \left| V_{1,i} - V_{2,i} \right|^m \right)^{\frac{1}{m}} \qquad (1.13)$$

*m* représente l'ordre est *W* représente la matrice de pondération.En faisant varier *m* on obtient différentes types de fonctions, la distance de Minkowski du premier ordre (*m=1*) est une distance de Manhattan et la distance de Minkowski du deuxième ordre (*m=2*) est une distance Euclidienne. Le choix d'une valeur appropriée pour *m* dépend de l'importance que nous voulons accorder aux plus grandes différences. Ainsi les grandes valeurs de *m* donnent progressivement plus d'importance aux différences les plus grandes et quand *m* tend vers l'infini la distance de Minkowski tend vers la distance de Tchebychev [18].

### 1.2.6.1.1 La distance Euclidienne

Lorsque *m=2* on obtient la distance euclidienne :

$$d(V_1, V_2) = \sqrt{\sum_{i=1}^{N} W_i \ X \ \left(V_{1,i} - V_{2,i}\right)^2} \qquad (1.14)$$

Elle est invariable aux translations et aux rotations des données dans les espaces des attributs, elle a été utilisée dans le système QBIC (Query by Image content) [19]. Cette métrique n'est pas invariante aux transformations linéaires et plus généralement à d'autres transformations qui dégradent les rapports entre les distances.

### 1.2.6.1.2 La distance de Manhattan

Pour m=1 on obtient la distance de Manhattan :

$$d(V_1, V_2) = \sum_{i=1}^{N} (\left|V_{1,i}, V_{2,i}\right|) \qquad (1.15)$$

Cette distance est aussi connue sous le nom de city-block. Elle est recommandée par le comité MPEG-7 [20] pour comparer deux formes décrites avec la méthode ART de Kim [21].

### 1.2.6.1.3 La distance de Tchebychev

Pour m=∞ on obtient la distance de Tchebychev :

$$d(V_1, V_2) = max_i\left(W_i\left|V_{1,i} - V_{2,i}\right|\right) \qquad (1.16)$$

Cette distance est adaptée aux données de grande dimension, elle est souvent employée dans les applications où la vitesse d'exécution est importante. Cette distance examine la différence absolue entre les différentes paires des vecteurs, elle est considérée comme une approximation de la distance Euclidienne mais avec moins de calcul.

### 1.2.6.1.4 La distance de Mahalanobis

La distance de Mahalanobis prend en compte la corrélation entre vecteurs au lieu des distances dans le cas quadratique [22]. La matrice *C* est la matrice de covariance entre *I* et *J* :

$$d_{Mah}(V_1, V_2) = \sqrt{(I - J).C^{-1}.(I - J)} \qquad (1.17)$$

### 1.2.6.1.5 La distance de Kullbak-Leibler

L'image est considérée comme une variable aléatoire dont les vecteurs d'attributs des pixels sont des realisations. La mesure de similarité se ramène à une mesure entre distributions de probabilités. L'équation (1.18) présente la distance de Kullbak-Leibler qui permet de mesurer la dissimilarité basée sur l'entropie mutuelle de deux distributions de probabilités [22]:

$$d_{Kull}(I, J) = \sum_{i=1}^{n} I_i \log \frac{I_i}{J_i} \qquad (1.18)$$

### 1.2.6.1.6 La distance de Jeffrey

Cette métrique est numériquement stable et fait preuve de plus de robustesse au bruit, elle est donnée par l'équation suivante [22]:

$$d_{Jeff}(I, J) = \sum_{i=1}^{n} I_i \log \frac{I_i}{m_i} + \sum_{i=1}^{n} J_i \log \frac{J_i}{m_i} \;\; avec \;\; m_i = \frac{I_i + J_i}{2} \qquad (1.19)$$

### 1.2.6.1.7 La distance de Bhatacharyya

La distance de Bhatacharayya [23] peut être utilisée pour comparer la similarité entre deux histogrammes *Q* et *V* de deux images, elle est définie par l'équation (1.20) :

$$d_{Bha}(Q,V) = 1 - \Sigma_i \sqrt{Q_i}\sqrt{V_i} \qquad (1.20)$$

### 1.2.6.1.8 La distance de Earth Mover

Cette distance a été utilisée dans les systèmes CBIR. Elle définie une mesure quantitative de travail minimale pour changer une signature en une autre. Le calcul de cette distance se ramène à la solution d'un problème de transport résolu par optimisation linéaire. Elles est alors définie comme [24] :

$$d_{EM}(I_1, I_2) = \frac{\sum_{i=1}^{n_{I_1}} \sum_{j=1}^{n_{I_2}} g_{ij} d_{ij}}{\sum_{i=1}^{n_{I_1}} \sum_{j=1}^{n_{I_2}} g_{ij}} \qquad (1.21)$$

Ou $d_{ij}$ indique la distance entre les composants et *I* et *J* et $g_{ij}$ est le flot optimal entre les deux distributions.

## 1.2.7 Recherche par similarité

Il s'agit de retrouver les vecteurs les plus similaires à un vecteur requête au sens d'une mesure de similarité donnée. La recherche par similarité peut être exécutée soit par une recherche des K plus proches voisins (k-ppv), soit par une recherche à ε près appelée également recherche par intervalle [25].

### 1.2.7.1 Recherche à ε prés

Il s'agit de rechercher l'ensemble des vecteurs qui se trouvent à une distance < ε du vecteur requête au sens de la mesure de similarité associée aux vecteurs. Cet ensemble est défini par :

*Q (q, ε)= { v ∈ BD / sim (q,v) < ε }* où *BD* est la base de vecteurs, *v* est un vecteur de la base, *q* est le vecteur requête et *sim* est la mesure de similarité associée aux vecteurs.

### 1.2.7.2 Recherche des K plus proches voisins

Il s'agit de chercher les K vecteurs les plus proches du vecteur requête.
Le résultat de la recherche est un ensemble de vecteurs défini par :

$$( \forall v_1 \in kppv) (\forall v_2 \in BD \ / \ kppv) \ sim(q,v_1) \leq sim (q,v_2).$$

Chacune de ces techniques de recherche possède des avantages et des inconvénients. La recherche des K plus proches voisins garantit systématiquement K vecteurs dans l'ensemble résultat. Cependant, certains de ces vecteurs peuvent être très éloignés du vecteur requête et ne peuvent être considérés comme similaires. La recherche à ε près permet à un utilisateur expérimenté qui maîtrise la distribution de ces vecteurs d'éviter ce problème en choisissant une valeur appropriée de ε. Mais dans le cas général, le choix de ε reste problématique : une trop petite valeur impose une recherche très restrictive et peut donner lieu à des ensembles résultats vides, une trop grande valeur peut engendrer à l'inverse, des ensembles résultats de très grande taille.

## 1.2.8 Exemples de systèmes de recherche d'images

De nombreux systèmes d'indexation et de recherche d'images par le contenu visuel ont vu le jour. Ces systèmes permettent de naviguer dans des bases d'images et d'exprimer des requêtes au travers d'interfaces conviviales pour la recherche d'images similaires. Parmi les systèmes les plus utilisés, nous pouvons citer :

### 1.2.8.1 Le système QBIC

QBIC (Query by image content) [26] est le premier système de recherche d'images commercial développé par la société IBM. Les descripteurs visuels qu'intègrent ce système sont la couleur, la texture et la forme. Pour l'attribut de couleur, sa signature correspond à un vecteur moyen 3D exprimé dans l'un des espaces de couleurs retenus : RGB, YIQ, Lab, etc. Pour la forme, elle correspond à un vecteur de quatre paramètres: la circularité de l'objet, l'excentricité, l'orientation principale et un ensemble de moments algébriques invariants. La distance employée pour comparer les images est la distance euclidienne et, pour les histogrammes,

QBIC intègre la distance quadratique. L'indexation multidimensionnelle appliquée dans QBIC pour accélérer la recherche est basée sur les arbres-R* [22].

### 1.2.8.2 Le système Virage

Virage est le moteur de recherche d'images développé par la société Virage Inc [27]. Son objectif est de construire un environnement dédié à la recherche d'images, principalement composé de primitives. Similairement à QBIC, virage propose des requêtes portant sur la couleur, la localisation des couleurs, la texture et la structure de l'image. L'interface de Virage offre la possibilité d'ajouter et de pondérer les différentes primitives ainsi que d'utiliser le bouclage de pertinence. L'avantage de Virage par rapport à QBIC est qu'il autorise une combinaison entre les différents modes de recherche. L'utilisateur définit le poids qu'il veut attribuer à chaque mode.

### 1.2.8.3 Le système Photobook

Photobook Photobook [28] est un système académique développé par le laboratoire MIT qui permet la navigation dans de grandes bases d'images en combinant l'annotation textuelle des images et les descripteurs extraits des images. La caractérisation des textures est basée sur l'approche « Wold decomposition » [29]. Les descripteurs retenus pour la texture sont la périodicité, l'orientation et grossièreté. Pour la forme, elle est modélisée par des éléments finis dont des attributs pertinents sont extraits. La mesure de similarité employée est la distance euclidienne [22].

### 1.2.8.4 Le système Netra

NeTra Le département d'électronique et de génie informatique de l'université de Californie à Santa Barbara a développé le système de recherche d'images NeTra [30] qui utilise une description par région segmentée dont chacune est caractérisée par la couleur, la texture, la forme et une localisation spatiale.Le premier attribut de couleur est représenté par un dictionnaire (code-book) avec une quantification sur 256 couleurs. Les filtres de Gabor et les ondelettes caractérisent la texture. Pour la forme, les descripteurs sont extraits des courbures et centroïde de la forme. La correspondance entre images s'effectuent par une distance euclidienne [22].

#### 1.2.8.5 Le système Blobword

Blobworld Blobworld [31], au même titre que Netra, repose sur la caractérisation des régions par la couleur, la forme, la texture et la localisation spatiale. La couleur dans le cas de Blobworld est représentée par un histogramme de 218 cases dans l'espace Lab. Le contraste et l'anisotropie caractérisent la texture. La surface, l'excentricité et l'orientation correspondent aux descripteurs de la forme. La requête d'un utilisateur consiste à choisir une région et à caractériser son importance par deux variables linguistiques (« somewhat » et « very »). La distance quadratique est employée pour la couleur et la distance euclidienne est utilisée pour la texture et la forme. L'indexation multidimensionnelle utilisée est similaire à celle de QBIC (arbre-R $*$ ) [22].

#### 1.2.8.6 Le système IKONA

IKONA [32] est un système de recherche et de navigation interactive dans de grandes bases de données multimédia développé par l'équipe IMEDIA de l'INRIA basé sur une architecture client/serveur. IKONA procède à l'extraction des descripteurs de la couleur à l'aide de l'histogramme de couleur pondéré [33]. L'extraction de la texture est effectuée en utilisant le spectre de Fourier. Pour la forme, il utilise l'histogramme d'orientation des contours [22].

## 1.2 Indexation et recherche de vidéo par le contenu

Les principaux modules de l'indexation des documents vidéos sont décrits par la figure 1.2. Le système comporte deux phases indissociables. Le premier concerne le mode *''présentation de vidéo''* et la seconde concerne *''l'utilisation de cette présentation''* dans un but de recherche .

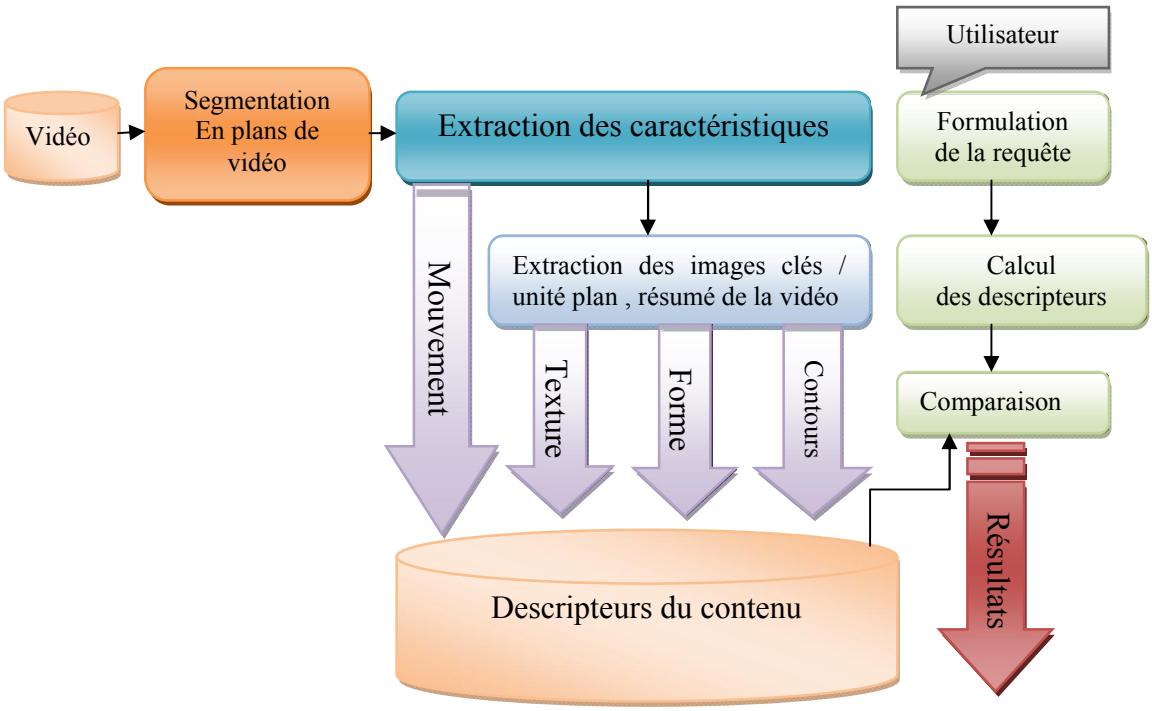


Figure 1.2 : principe d'indexation de vidéo par le contenu

## 1.3.1 Segmentation en plan de vidéo

Il s'agit de découper une vidéo pour avoir une séquence d'images individuelles et à partir de ces images, nous pouvons définir la suite des unités de base appelées ''plans''. Chaque plan est identifié par une image clé [34] et contient un ensemble d'images similaires. L'ensemble de ces images (Figure 1.3) forme ce que l'on appelle le ''résumé vidéo''.

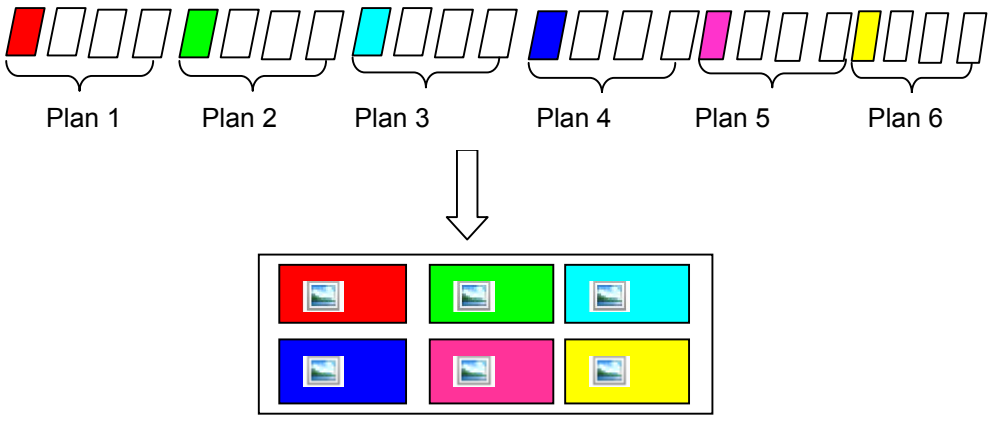


Figure 1.3 : Les plans, les images clefs et le résumé vidéo

## 1.3.2 Description de changement de plans

Un plan vidéo est le regroupement des images successives en des unités qui correspondent à des prises de vues de la caméra. Le plan est souvent l'unité temporelle la plus petite pour une séquence vidéo si l'on ne prend pas en compte l'image pour laquelle la notion du temps a disparu [35]. Chaque plan est séparé du précédent et du suivant par une transition. On distingue deux types de transitions [36] :

a- les transitions brusques : la dernière image du plan en cours est directement suivie par la première image du plan suivant sans aucun effet inséré entre les deux plans. La figure 1.4 donne un exemple de brusque transition.


Figure 1.4 Exemple d'une transition brusque

b- Les transitions progressives : c'est le cas où les deux plans successifs sont connectés en utilisant un effet particulier. De ce fait, plusieurs transitions peuvent se présenter dont le type le plus connu est le type fondu. La figure 1.5 donne un exemple de transition progressive.

Figure 1.5 Exemple d'une transition progressive

### 1.3.3 Segmentation en scène

La scène est une unité sémantique importante puisqu'elle contient une séquence de plans dont la logique permet d'exprimer une idée. Par contre, le plan est une unité technique qui est souvent de courte durée et son étude isolée ne permet pas de comprendre réellement le déroulement de la scène. La détermination des scènes est alors utile pour la navigation, la visualisation des données audiovisuelles et aussi pour leur analyse sémantique. Elles permettent aux utilisateurs d'avoir une bonne idée de l'action s'y déroulant ou de l'ambiance dégagée. La première étape incontournable est le découpage en plans, ensuite l'étude de l'organisation des plans permet de grouper les plans en scènes [34]. Par ailleurs le contenu d'une scène peut-être visuellement très varié et la notion de plan est toujours nécessaire pour identifier et caractériser les différents contenus. L'indexation des scènes repose donc sur les plans et plus particulièrement leurs images caractéristiques.

### 1.3.4 L'image représentative

Le processus de résumé vidéo consiste à sélectionner une seule image parmi toutes les images pour chaque plan obtenu par la segmentation en plan de la vidéo. Cette image connue souvent par le nom image représentative ou image clef, elle est caractérisée par la richesse en terme d'information. Cependant toutes les images qui composent un plan vidéo ayant une forte similarité entre elles, de ce fait les approches empiriques sélectionnent simplement la première, la dernière ou l'image médiane du plan.

### 1.3.5 Les techniques de segmentation en plan de la vidéo

Chacune de ces méthodes présente des avantages et des inconvénients et la plupart d'entres elles utilisent des seuils fixés de manière empirique. Elles se basent sur le principe de calcul de distances de similarité entre les images successives permettant ainsi de construire des groupes d'images appelés plan. Le calcul d'une mesure de similarité entre deux images successives d'une séquence vidéo [37].

Ces méthodes peuvent être basées sur la différence pixel à pixel, sur la différence d'histogrammes, sur la différence de mouvement, sur la différence de

blocs, sur les distances entre les structures arborescentes associées aux images de la séquence vidéo ou sur la différence entre les descripteurs visuels.

### 1.3.5.1 Différences pixel à pixel

Les premières méthodes basées sur les différences pixel à pixel, elles détectent un changement de plan en calculant une différence entre les pixels de l'image à l'instant *t* et ceux de l'image à l'instant *t+1*. Ces méthodes comparent deux images successives en utilisant différentes mesures plus ou moins robustes qui ont été progressivement proposées tout le long des recherches, telles que la différence de l'énergie (mesurée par la somme des carrés des intensités) des deux images.
Il est également possible de déterminer, dans un premier temps, pour chaque couple de pixels appartenant aux deux images si leurs intensités sont proches ou non. La mesure de différence est alors obtenue, dans un second temps, en comptant le nombre de couples ayant des intensités éloignées.
Si le nombre de pixels $N_{pixels}$ qui change d'une image à l'autre dépasse un certain seuil (T) un changement de plan est détecté. Le nombre de changements de pixels peut être mathématiquement calculé par la formule suivante [38]:

$$N_{pixels} = \frac{\sum_{x,y=1}^{X,Y} D(x,y,t,t+1)}{X.Y} \qquad (1.22)$$

Tel que :

$$D(x,y,t,t+1) = \begin{cases} 1 & si\ |D(x,y,t,t+1) - D(x,y,t,t+1)| > T \\ 0 & si\ non \end{cases} \qquad (1.23)$$

Y ,X correspondent aux formats d'image horizontaux et verticaux, et T est un seuil spécifiant le départ minimum de valeur absolue de la différence à partir de laquelle un pixel est considéré comme étant modifié.

### 1.3.5.2 Différence d'histogrammes

Afin de pallier au manque de robustesse des méthodes basées sur les pixels, certains auteurs ont proposé les méthodes à base d'histogramme. Ces méthodes effectuent une comparaison entre deux images successives en s'appuyant sur leurs

histogrammes respectifs donc elles utilisent des statistiques plus globales, les histogrammes.

Un contenu de couleur peut être décrit de manière efficace à l'aide d'histogrammes de couleurs. Le principe peut être énoncé comme suit [38]:

- ✓ Décrire chaque image vidéo à l'aide d'un histogramme de couleur.
- ✓ Évaluer la variation entre chacune des deux trames consécutives en tant que distance / similarité en utilisant leurs histogrammes couleur associés.
- ✓ identifier le changement de plan chaque fois que la mesure évaluée à l'étape précédente est supérieure à un seuil donné.

Cette différence peut être calculée en considérant les images en niveaux de gris ou en couleur, dans un espace relativement limité (par exemple 2 bits pour chaque composante couleur) ou dans l'espace original.

La méthode basée sur histogrammes de niveaux de gris en calculant une distance entre les histogrammes des images comme le montre l'équation suivante [39] :

$$\left(\sum_{v=0}^{v} |H(I_t, v) - H(I_{t-1}, v)|\right) > T \qquad (1.24)$$

Dans le cas d'images couleur, la mesure de distance entre deux images peut être calculée comme la plus grande des différences entre deux histogrammes de même couleur pour cela une méthode similaire est proposée en utilisant seulement 64 bins pour les histogrammes de couleurs (2 bits pour chaque composante de couleur de l'espace RGB). Pour la détection de changement de plan la notation suivante $H_{64}(I,v)$ est utilisée dans l'équation suivante [40]:

$$\left(\sum_{v=0}^{64} |H_{64}(I_t, v) - H_{64}(I_{t-1}, v)|\right) > \mathrm{T} \qquad (1.25)$$

D'autres mesures de distance peuvent être utilisées, telles que la mesure de similarité du cosinus, la distance quadratique, l'entropie croisée, la divergence. Il est possible de calculer l'intersection entre deux histogrammes de deux images successives. Plus celle-ci sera faible, plus la probabilité d'être en présence d'un changement de plans sera important.

### 1.3.5.3 Estimations de mouvement

Les méthodes basées sur une estimation de mouvement utilisent l'information de mouvement comme critère principal pour la détection des changements de plans. Les mouvements sont estimés pour chaque pixel d'une image obtenue à l'instant *t*, et sont comparés avec ceux de l'image correspondant à l'instant *t+1*. Un nombre trop important de mouvements incohérents entre les deux images successives implique alors la détection d'un changement de plan.

### 1.3.5.4 Les blocs

Les méthodes basées sur les blocs sont des méthodes intermédiaires entre les méthodes basées pixels (locales) et les méthodes basées histogrammes (globales). L'analyse des groupes d'images successives, dans la détection des changements de plans, peut aussi être effectuée à un niveau intermédiaire entre le niveau local (les pixels) et le niveau global (les histogrammes). Les méthodes entrant dans cette catégorie analysent l'image par blocs. De cette façon, il est possible de comparer deux images en mesurant les différences pour chaque paire de blocs via un calcul de la moyenne et de l'écart-type des intensités pour chaque bloc. Une fois chaque paire de blocs étiquetée selon leur similarité, la détection dépend du nombre de paires de blocs différents [36].

### 1.3.5.5 Segmentation temporelle de la vidéo par les descripteurs MPEG7

Dans cette méthode le descripteur histogramme de contours (Edge Histogram Descriptor : EHD) était proposé pour la détection de changement de plans de la vidéo [41]. La méthode d'application du EHD consiste à extraire ce descripteur pour chaque image de la séquence vidéo et à comparer les vecteurs des caractéristiques de chaque deux images consécutives. La mesure de degré de similarité se base sur la caractéristique visuelle donnée par le descripteur EHD. Si cette distance dépasse un certain seuil l'existence d'un changement de plan est signalé.

### 1.3.5.6 Combinaison des méthodes

Comme son nom l'indique, c'est une famille des méthodes qu'en combinant différentes approches plus simples, comme par exemple, une approche basée sur les pixels et une approche basée sur les histogrammes [36]. Elles sont caractérisées par un temps de calcul important. L'intérêt de ces méthodes réside dans le fait que les différentes techniques sont soit exécutées successivement avec des seuils de tolérance de plus en plus élevés, soit simultanément en définissant une règle de fusion des décisions obtenues, par exemple un "ET" ou un "OU".

### 1.3.5.7 Segmentation par la méthode d'arbre R

L'arbre R (ou Rtree) est une structure hiérarchique utilisée pour indexer les objets spatiaux ou géométriques [37]. Les objets spatiaux sont représentés par des rectangles minimums englobant les données (REM) sur une image [42]. Cette technique se déroule en trois étapes :

- *Étape 1* : découper la vidéo en séquence d'images individuelles par des algorithmes ou technique de programmation ainsi que des logiciels de traitement de la vidéo.
- *Étape 2* : segmenter et construire l'arbre R pour chaque image obtenue lors de l'étape1.
- *Étape 3* : calculer les distances de R-Similarité entre les arbres représentatifs des images obtenues dans l'étape précédente. Selon un seuil calculé d'une manière empirique, on détectera le changement de plans de la vidéo.

## 1.4 Conclusion

Le domaine de l'indexation et de la recherche par le contenu que ce soit dans une base d'images ou de vidéos est un domaine actif et très vaste. Très souvent, les systèmes de recherche d'images et de vidéos par le contenu s'appuient sur la description visuelle de bas niveau relatifs à la couleur, la texture et la forme. Les travaux réalisés ont prouvé que la réalisation d'un système d'indexation et de recherche par le contenu, dans lequel l'utilisateur a la possibilité d'effectuer des recherches fines au sein d'une base de données multimédias, induit de gros efforts et un grand travail.

Nous avons présenté dans ce chapitre, les notions de base sur l'indexation et la recherche par le contenu. Un certain nombre de méthodes de segmentation en plans de vidéo sont présentées ; ceci nous a donné une vue globale sur les techniques de détection de changement de plans. La technique utilisant la structure d'arbre -R nous a permis d'envisager la possibilité d'utiliser d'autres structures arborescentes dans ce domaine. Le chapitre suivant sera consacré à l'étude détaillée des travaux relatifs aux structures arborescentes dans le domaine de l'indexation et la recherche par le contenu.

.

# CHAPITRE 2
# STRUCTURE D'ARBRE POUR L'INDEXATION

## 2.1 Introduction

Les techniques d'indexation d'images ont pour but d'organiser un ensemble de descripteurs afin que les procédures de recherche soient performantes en temps de réponse. Cette organisation se traduit généralement par une structuration des descripteurs en petits ensembles et par l'application de stratégies de recherche capable de filtrer toutes les images non pertinentes qui seront évitées pendant la recherche garantissant ainsi un temps de recherche acceptable par l'utilisateur.

La technique d'Indexation en système de la recherche d'image par similitude visuelle se divise en deux étapes [5] : **1)** la première consiste à extraire des caractéristiques visuelles de l'image. Des méthodes d'analyse permettent ainsi d'extraire de l'image un résumé de son apparence visuelle correspondant au descripteur d'image. Après extraction des caractéristiques visuelles, le contenu visuel de l'image est décrit par un vecteur numérique (ou un ensemble de vecteurs), que l'on appelle généralement signature de l'image. La mise en place d'un descripteur consiste également à associer aux signatures une mesure de similarité. Au moment de la recherche, deux images seront jugées similaires si leurs signatures sont similaires au sens de cette mesure de similarité. **2)** la deuxième étape consiste à structurer l'espace des signatures obtenu précédemment en l'organisant selon une structure d'index (généralement multidimensionnel) permettant ainsi d'accélérer la recherche. Les approches les plus récentes s'attachent également à réduire les coûts en temps CPU correspondant généralement au calcul de la mesure de dissimilarité. La recherche via un index doit naturellement être plus rapide qu'un parcours exhaustif de l'espace.

L'objectif de ce chapitre est de présenter un état de l'art sur les principales méthodes d'arbres utilisées en indexation pour la structuration des vecteurs descripteurs

de la base d'images ou pour la structuration de l'espace de données. Nous présentons d'abord le principe général de chaque méthode en indiquant les insuffisances et les points forts de chacune des techniques.

Ensuite nous présentons le problème de la malédiction de la dimension ; la structure d'index (arbre quaternaire) sera présentée comme une structure arborescente non équilibrée. En fin, nous terminerons par une conclusion.

## 2.2 Méthodes d'arbre en indexation multidimensionnelle

Les techniques principales de la méthode d'arbre en indexations multidimensionnelles visent à regrouper les descripteurs de base et à les englober dans des cellules faciles à manipuler (hiérarchie). Les index multidimensionnels nous permettent d'éviter de considérer tous les descripteurs dans la base lors d'une recherche en considérant seulement les groupes ou les paquets les plus pertinents ; par la suite, seuls les descripteurs dans les paquets sélectionnés seront considérés.
Il y a deux grandes catégories de techniques de création de cellules ; celles qui procèdent par partitionnement des données et celles qui partitionnent l'espace [3].

### 2.2.1 Partitionnement des Données

Les techniques de partitionnement de données créent des cellules en se basant sur la distribution des descripteurs et leur proximité relative dans l'espace. Dans cette catégorie, on trouve des techniques comme la famille R-tree [4-43-44], SS-tree[45], SR-tree [46] et X-tree [47].

#### 2.2.1.1 La Famille R-tree : Rectangle-Rree

Dans la famille R-tree, on trouve trois structures R-tree [4], R+-tree[43] et R*tree[44]. L'idée de base de cette approche est l'indexation des objets spatiaux par des rectangles englobants minimum. Dans le contexte de l'indexation multidimensionnelle, on utilise des rectangles englobants multidimensionnels (hyper-rectangle).
Le principe de cette famille est la hiérarchie d'hyper- rectangles englobants et non-disjoints correspondant à la distribution des données par un arbre équilibré, les données étant au niveau des feuilles. La figure 2.1 présente un exemple de la structure R-Tree.

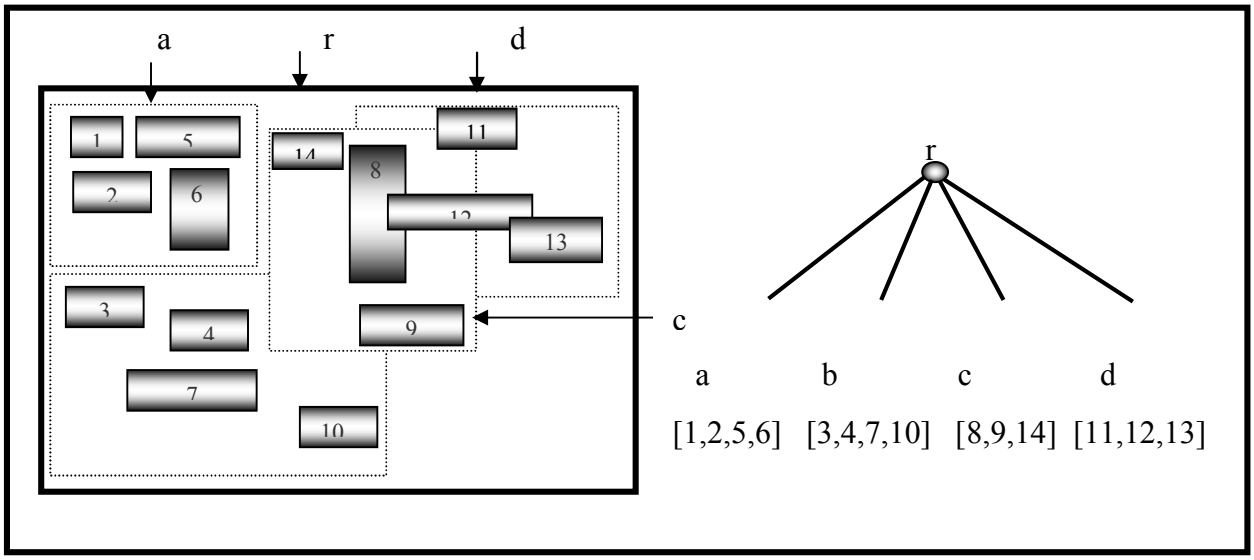


Figure 2.1: Structure du R-Tree

Un arbre R-tree a les propriétés suivantes [4] :

• Au niveau des feuilles, le rectangle englobant est le rectangle minimum recouvrant les vecteurs de données appartenant à ce nœud. Chaque nœud feuille contient au maximum *M* et au minimum $m \leq M/2$ éléments de données.

• Tous les nœuds, sauf la racine, qui ne sont pas des feuilles ont entre *m* et *M* nœuds fils, le rectangle englobant de ces nœuds est le rectangle minimum qui englobe les rectangles englobants des nœuds fils.

• Le nœud racine a au moins deux fils sauf quand il est une feuille. Le rectangle englobant du nœud racine recouvre tous les vecteurs de données de la base.

• Toutes les feuilles sont au même niveau.

Un rectangle englobant est déterminé par deux points $S(s_1,s_2,\ldots,s_n)$ et $T(t_1,t_2,\ldots,t_n)$ ; pour chaque élément $X(x_1,x_2,\ldots,x_n)$ appartenant à ce rectangle, on a :

$$s_i \leq x_i \leq t_i \ , \ \forall i \in [1,n]$$

Pour rechercher toutes les données appartenant à un rectangle *Q* quelconque, on réalise la recherche à partir du nœud racine, en descendant aux nœuds fils qui ont le rectangle englobant intersectant le rectangle et ainsi de suite jusqu'à ce qu'on rencontre les nœuds feuilles. Au niveau des nœuds feuilles, on retourne toutes les données appartenant au rectangle *Q*.

La création d'un arbre R-tree est réalisée en ajoutant au fur et à mesure des vecteurs dans l'arbre. L'algorithme exact pour l'insertion d'un nouveau élément (vecteur) dans l'arbre est dans [4], l'idée principale est de partir du nœud racine, puis de descendre pour trouver la feuille où ajouter le nouveau élément. A chaque nœud, on choisit le nœud fils avec le rectangle englobant le plus petit. Lorsqu'on trouve une feuille pour ajouter l'élément, s'il est plein, on doit le diviser en deux feuilles différentes en minimisant la surface totale des deux nouveaux rectangles englobants (division exhaustive, coût quadratique ou linéaire [4]). Le rectangle englobant de chaque nœud est créé et mis à jour au cours de l'insertion des éléments dans l'arbre pour qu'il soit le rectangle le plus petit possible qui recouvre tous les éléments dans le sous-arbre de ce noeud.

La structure R-tree est dédiée à la recherche par intervalles, elle convient pour l'indexation des données spatiales. Lors de la recherche, on peut gagner du temps, car on ne doit pas considérer tous les éléments dans la base, on considère seulement les nœuds fils ayant le rectangle englobant qui intersecte l'intervalle entré. Mais la possibilité d'intersection entre les rectangles augmente quand le nombre de dimensions est grand. Dans ce cas, le parcours séquentiel peut être plus efficace.

Les structures R+-tree [43] et R*-tree [44] sont données pour optimiser la recherche en minimisant le chevauchement des rectangles englobants. La structure R+-tree évite le recouvrement entres les rectangles en divisant chaque rectangle qui recouvre un autre rectangle en plus petits rectangles jusqu'à ce qu'il n'y arra plus de recouvrement. Cela peut faire augmenter la hauteur de l'arbre, mais on peut gagner du temps en réduisant le nombre de sous-arbres à visiter. La structure R*-tree minimise le recouvrement entre les rectangles et minimise aussi le volume des rectangles en appliquant, quand on veut ajouter un nouveau fils à un nœud plein, le mécanisme de réinsérer quelques nœuds fils du nœud plein avant de le diviser en deux avec l'espoir de trouver de meilleures positions pour les nœuds à réinsérer. R*-tree est la structure ayant le plus de succès dans la famille R-tree [44-46]. Les expérimentations [44] montrent qu'un R*-tree peut être utilisé efficacement pour l'organisation des données multidimensionnelles et des données spatiales.

### 2.2.1.2 SS-tree : Similarity Search Tree

Le SS-tree [45] est une structure d'indexation de similarité qui regroupe les vecteurs de caractéristiques suivant la similarité entre eux. La mesure de similarité utilisée ici est la distance euclidienne dans le cas où les poids de toutes les dimensions dans le vecteur de caractéristiques sont pareils. La structure de SS-tree ressemble à celle du R-tree mais on remplace dans chaque nœud le rectangle englobant par une sphère englobante représentée par un centre et un rayon. Les données sont toujours au niveau des feuilles. Le centre de la sphère englobante d'un nœud est le centre de gravité de tous les éléments dans le sous-arbre de ce nœud. Au niveau des feuilles, la sphère englobante recouvre les éléments appartenant à cette feuille, le rayon de la sphère englobante est égal à la distance entre le centre et le point le plus loin. Et au niveau d'un nœud interne, la sphère englobante recouvre les sphères englobantes de tous les nœuds fils de ce nœud, le rayon de la sphère est toujours supérieur ou égal à la distance entre le centre et le point le plus loin parmi les points dans le sous arbre de ce nœud.

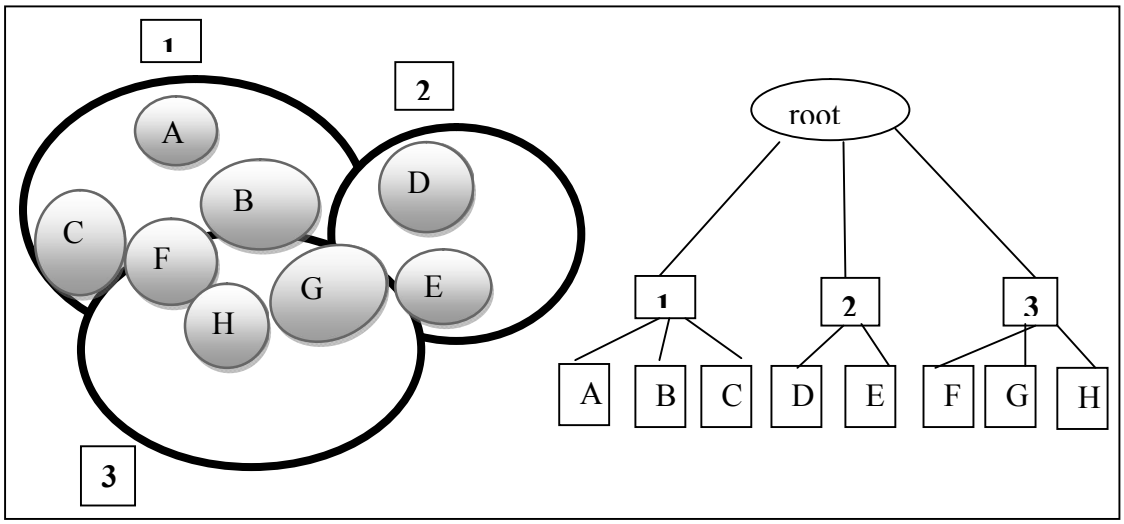


Figure 2.2 : Structure du SS-Tree

On utilise aussi le mécanisme de réinsertion d'une partie d'un nœud plein comme dans le cas du R*-tree afin de minimiser le recouvrement entre les sphères englobantes et le volume des sphères. Les expérimentations [45] montrent que le SS-tree est meilleur que le R*-tree pour les applications de recherche de similarité avec des données de haute dimensionnalité.

### 2.2.1.3 SR-tree : Sphere/Rectangle Tree (1997)

L'idée du SR-tree (Sphere/Rectangle Tree) est de combiner les deux structures R*-tree et SS-tree en identifiant la région de chaque nœud par l'intersection du rectangle englobant et de la sphère englobante [46]. Pour chaque nœud dans l'arbre SR-tree, on détermine :

• Un rectangle englobant recouvre tous les éléments (vecteurs) dans le sous-arbre de ce nœud. Ce rectangle est déterminé comme dans le cas du R-tree.

• Une sphère englobante recouvre tous les éléments dans le sous-arbre de ce nœud. Cette sphère est déterminée comme dans le cas du SS-tree. Un élément (vecteur) pourra être inclus dans le sous-arbre d'un nœud quelconque s'il appartient à la région déterminée par l'intersection du rectangle englobant et de la sphère englobante de ce nœud.

La structure SR-tree est basée en même temps sur la structure de la famille R-tree [4-43-44] et celle de SS-tree [45]. Elle regroupe des avantages de la famille R-tree et de SStree en déterminant une région d'un nœud par l'intersection du rectangle englobant minimal et de la sphère englobante minimale des données dans le sous-arbre de ce nœud.

Pourquoi combiner les hypercubes et les hypersphères ? Dans la stratégie de création du R-tree, on essaie de minimiser le volume des hyper-rectangles. Mais dans certains cas, un hypercube peut avoir une plus grande diagonale que celle d'un hypercube ayant un plus petit volume. Les rectangles de grande diagonale peuvent provoquer plus de recouvrement entre des nœuds. En créant le SS-tree, on essaie de minimiser le diamètre des hypersphères tandis que les hypersphères occupent un grand volume pour un petit diamètre dans le cas de grande dimensionnalité. L'intersection du rectangle englobant et de la sphère englobante permet d'obtenir une région plus petite que celles du R-tree et du SS-tree. D'ailleurs, dans la stratégie pour minimiser le diamètre, l'intersection donne des régions de petits volumes pour un petit diamètre, et donc, permet de réduire la superposition des régions.

Un arbre SR-tree a les propriétés suivantes :

• Les données sont au niveau des feuilles. C'est-à-dire que le rectangle englobant et la sphère englobante d'un nœud feuille sont respectivement le rectangle minimal et la sphère minimale recouvrant les vecteurs de données appartenant à ce nœud. Chaque nœud feuille contient au maximum $M_L$ et au minimum $m_L \leq M_L/2$ éléments de données.

• Tous les nœuds qui ne sont pas les feuilles sauf la racine ont entre $m_N$ et $M_N$ nœuds fils ; le rectangle englobant et la sphère englobante de ces nœuds recouvrent respectivement les rectangles englobants et les sphères englobantes des nœuds fils.

• Le nœud racine a au moins deux fils sauf quand il est une feuille. Le rectangle englobant et la sphère englobante du nœud racine recouvre tous les vecteurs de données de la base.

• Toutes les feuilles sont au même niveau.

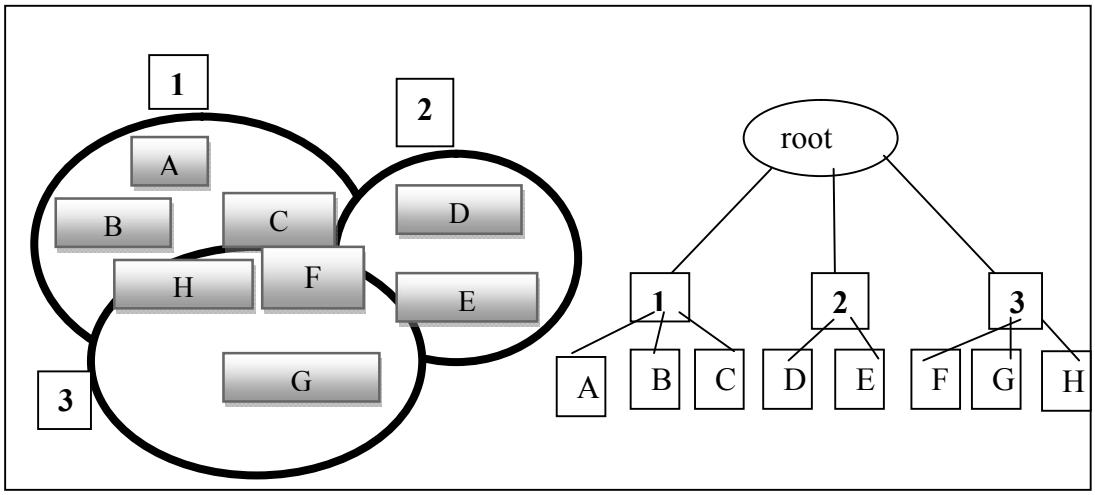


Figure 2.3 : Structure du SR-Tree

En combinant le rectangle englobant et la sphère englobante (ou plus précisément l'hypercube et l'hypersphère dans le contexte de données de grande dimension), on garde l'avantage du comportement en grande dimension du SS-tree et on peut diminuer le recouvrement entre les nœuds par rapport aux cas du R*-tree et du SS-tree en créant des régions de petits volumes et de petits diamètres [46]. Cela accroît la performance des requêtes des plus proches voisins pour les données de haute dimensionnalité [46]. Cela implique aussi que cette structure est utile pour les applications d'indexation de similarité des images/vidéos.

### 2.2.1.4 X-Tree

Le X-tree est une variance de la structure R*-tree [47]. Elle améliore la performance du R*-tree en utilisant l'algorithme de partitionnement minimisant le recouvrement et le mécanisme des super-noeuds qui ont plus de nœuds que les nœuds normaux [46-47]. Les super-noeuds sont utilisés si le recouvrement généré quand on ajoute un nouveau fils dans un nœud plein est plus petit que celui lorsqu'on divise ce nœud en deux nouveaux nœuds. Les expérimentations [46-47] montrent que la

performance du X-tree pour les requêtes par point (point query) est meilleure que celle du R*-tree.La figure 2.4 présente un exemple de X-Tree.

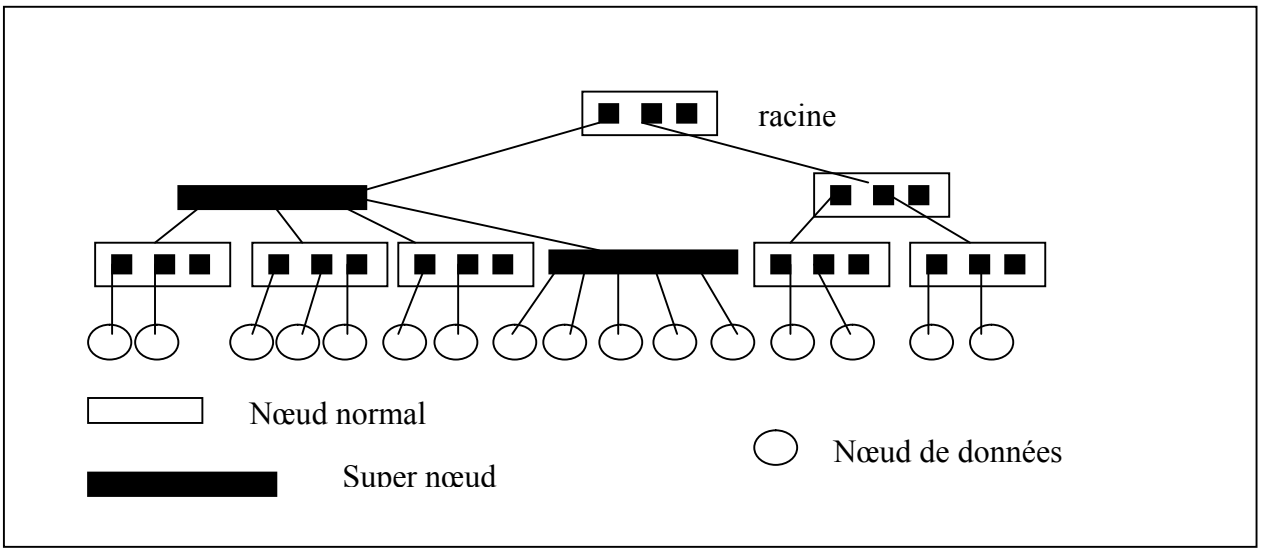


Figure 2.4 : Structure du X-Tree

## 2.2.2 Partitionnement de L'espace

À l'égard des techniques de partitionnement de l'espace, on divise directement l'espace multidimensionnel en cellules plus ou moins complexes et régulières. Quelques techniques de ce type sont KD-tree [48], KDB-tree [49], LSD-tree [50].

### 2.2.2.1 KD-Tree

Le KD-tree (k-dimensional tree) est une methode d'indexation multidimentionnelle permettant d'organiser les données dans un espace multidimentionnel de dimenssion *d* [48], c'est une structure arborescente binaire baseé sur une division récursive de l'espace en région hyper-rectangulaire disjointes, appelleés cellules. Chaque nœud dans l'arbre est asscoié à une region ainsi qu'a l'ensemble des vecteurs se trouvant dans cette région. Le nœud racine de l'arbre est associé au cadre limite qui contient tous les vecteutrs, les nœuds internes contiennent l'yperplan séparateur en plus des deux fils correspondant aux deux sous espaces génerées par le partitionnent. Enfin les feuilles de l'arbre contiennent l'ensemble des vecteurs dans le sous espace correspondant. Plusieurs algorithmes de partitionnement exitent, ils différent dans le choix du plan séparateur, et dans la condition d'arret de l'algorithme [3]. La figure 2.5 illustre un exemlpe de la structure KD-Tree.

.

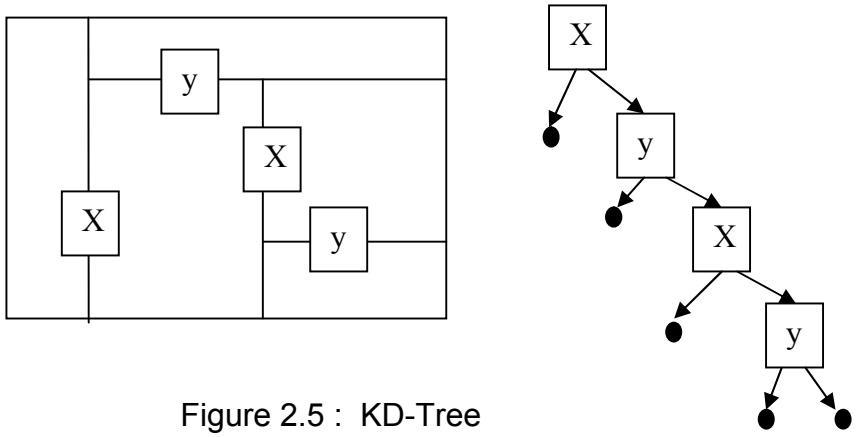


Figure 2.5 : KD-Tree

Un arbre KD-tree a des propriétés suivantes :

• La racine est la boîte englobante de tout l'espace.

• Un nœud correspond à un plan séparateur et deux pointeurs pointant vers deux sous-espaces construits par le plan.

• Une feuille correspond à la liste des objets de la base appartenant à l'espace de ce nœud.

Cette structure permet la recherche par intervalles ou par plus proches voisins. Elle permet d'accélérer le traitement des données multidimensionnelles. L'avantage de cette structure pour la recherche dépend de la concentration des données dans l'espace.

### 2.2.2.2 KDB-Tree

Le KDB-tree [49] est une structure d'indexation pour les vecteurs multidimensionnels. Elle combine les propriétés du KD-tree et du B-tree en organisant les données sous forme d'un arbre balancé qui partitionne à chaque niveau l'espace de recherche en deux sous-espaces successivement selon chaque dimension. La disjonction entre les nœuds de même niveau dans l'arbre KDB-tree implique un seul chemin dans la recherche par point (point query). Mais dans le KDB-tree, lorsqu'on divise une région d'un nœud intermédiaire selon un axe, on doit aussi diviser les régions des sous-nœuds selon cet axe, cela pouvant créer des nœuds vides ou presque vides. Cela implique une diminution de la performance du KDB-tree dans le cas des requêtes par intervalles ou les requêtes par plus proches voisins.

### 2.2.2.3 LSD-tree : Local Split Decision Tree

Le LSD-tree (Local Split Decision tree) [50] est une technique d'indexation trés similaire au KD-Tree. Le LSD-Tree est un arbre binaire dans lequel chaque nœud représente un partitionnnement de l'espace en deux à l'aide d'un hyperplan. Ainsi, un partionnement est représenté par le numéro de la dimention selon laquelle le partionnement est effectué et une position sur l'axe associé à cette dimension. Ces deux informations sont stockées au niveau de chaque nœud. Au niveau le plus bas de l'arbre on trouve les pages de données. La figure 2.6 donne un exemple pour la structure d'arbre LSD-Tree.

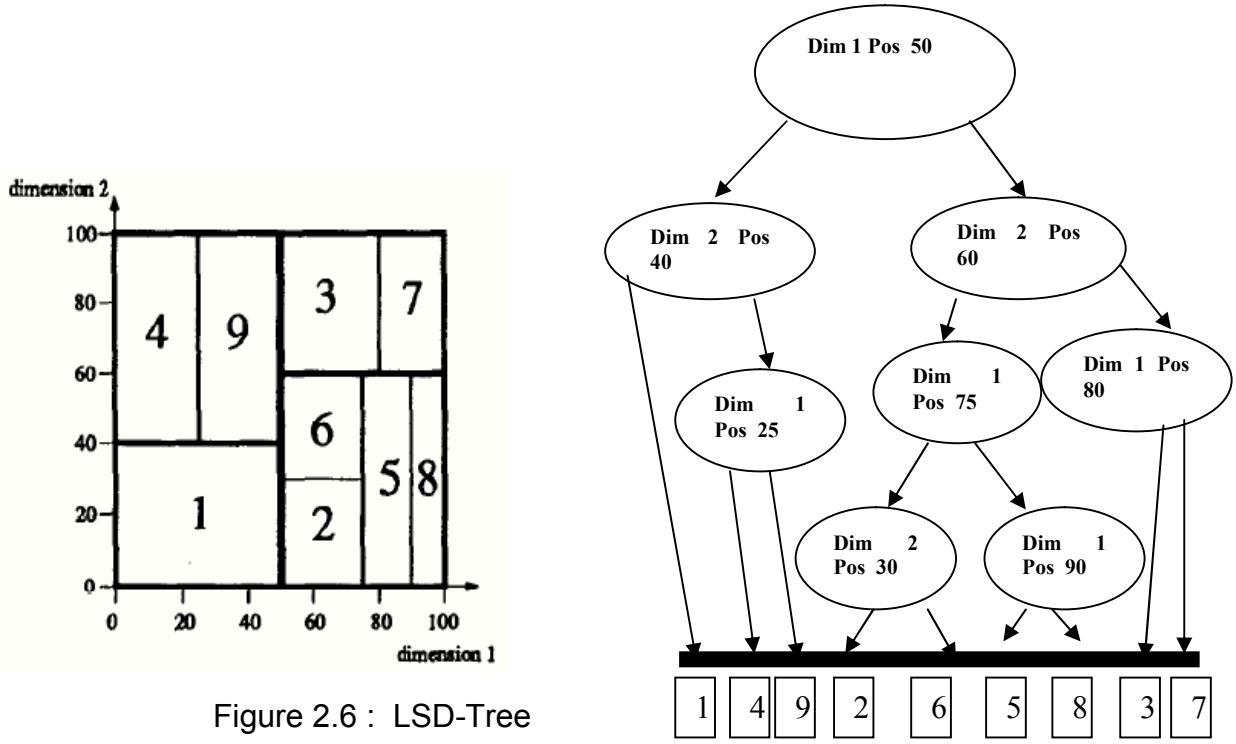


Figure 2.6 : LSD-Tree

Synthèse : le tableau 2.1 récapitule les avantages et inconvénients des différentes méthodes présentées précédemment. Un état de l'art complet sur ces structures a été fait par Gaede et al. [51] :

| Méthode | Avantages | Inconvénients |
|---|---|---|
| R*-Tree | Formes englobantes permettant d'affiner le filtrage | Fort taux de chevauchement; éclatement des nœuds problématiques |
| X-Tree | Limite le chevauchement entre régions | Paramètres de construction difficile à fixer : seuil max de chevauchement |
| SS-Tree | Petites tailles d'index | Chevauchement important |
| SR-Tree | Formes englobantes plus adaptées A de grandes dimensions | Complexité des formes; recherche coûteuse; taille d'index importante |
| Kdb-Tree | Pas de chevauchement | Fractionnements récursifs coûteux et arbitraires; faible taux d'utilisation de l'espace alloué |
| LSD-Tree | Codage binaire des régions: Taille de l'arbre réduite et zones vides non gérées | Organisation mémoire complexe |

Tableau 2.1 : Les avantages et inconvénients des différentes méthodes [51]

Le tableau ci-dessous résume les avantages et les inconvénients des index multidimensionnels cités auparavant [25] :

| **Index** | **Forme englobante** | **complexité** | **équilibre** | **chevauchement** |
|---|---|---|---|---|
| R-Tree | hyper rectangle | Calculable | oui | Oui |
| SS-Tree | hyper sphère | Calculable | oui | Oui |
| SR-Tree | ∩hyper rectangle et hyper sphère | Calculable | oui | Oui |
| KDB-Tree | hyper rectangle | Calculable | oui | Non |
| LSD-Tree | hyper rectangles | non calculable | non | Non |

Tableau 2.2 : les avantages et des inconvénients des différentes méthodes [25]

Les insuffisances ainsi que les points forts des différentes méthodes d'indexation sont donnés dans le tableau 2.3.

| Index | Points faibles | Points forts |
|---|---|---|
| R-Tree | Forme englobante avec une grande diagonale => processus de recherche dégradé; taille importante des nœuds; fort taux de chevauchement. | Forme englobante permettant d'affiner les règles de filtrage |
| SS-Tree | Forme englobante avec un grand volume => le taux de chevauchement élevé. | Forme englobante avec une faible diagonale => arbre compact. |
| SR-Tree<br>. | Formes englobantes complexes. | Forme englobante adaptée aux grandes dimensions |
| KDB-Tree | Faible taux d'utilisation de l'espace alloué | Pas de chevauchement |
| LSD-Tree | Recherche non précise des données | Bonne utilisation de l'espace alloué. |

Tableau 2.3. Récapitulatif point faibles et forts des méthodes

## 2.3 Structure d'index arborescente non équilibrée: Les arbres quaternaires

L'arbre quaternaire est une structure de données qui permet de représenter les images à deux dimensions. Elle est basée sur la décomposition récursive [52] de l'image en quadrants réguliers selon un critère particulier telle que l'homogénéité de la couleur des pixels ou homogénéité de la texture. L'arbre quaternaire est une structure hiérarchique construite par divisions récursives de l'espace en quatre quadrants disjoints [53]. Cette structure est très utilisée pour représenter les images, c'est-à-dire pour stocker les images elles-mêmes [54] ou pour stocker et indexer les caractéristiques des images [54-55-56]. En particulier, l'arbre quaternaire permet de tenir compte, lors de la recherche d'images similaires, de la localisation spatiale des caractéristiques d'images, telles que la couleur, le contour ou de la localisation des objets d'intérêt.

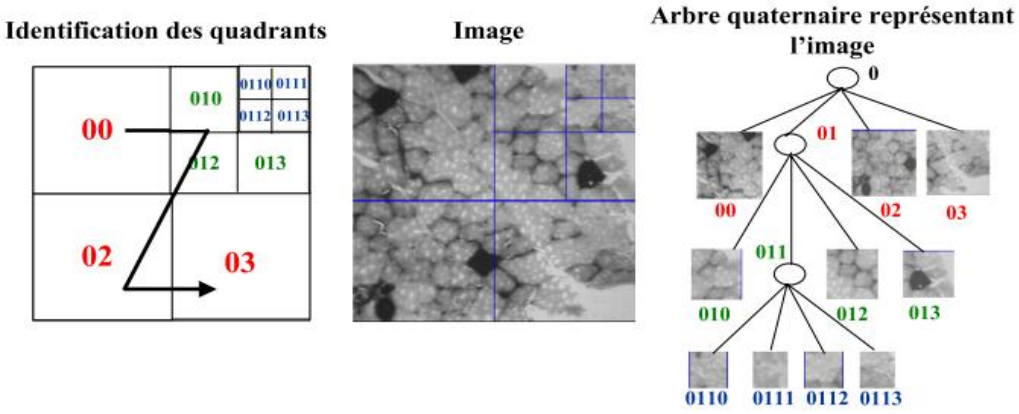


Figure 2.7 : Exemple d'arbre quaternaire

Pour être représentée par un arbre quaternaire, une image est récursivement décomptée en quatre quadrants disjoints de même taille, en fonction d'un critère de découpage de telle sorte que chaque nœud de l'arbre quaternaire représente un quadrant dans l'image. Le nœud racine de l'arbre représente l'image entière. Si une image n'est pas homogène par rapport au critère de découpage, le nœud racine de l'arbre quaternaire représentant l'image à quatre nœuds fils représentant les quatre premiers quadrants de l'image. Un nœud est feuille si le quadrant correspondant dans l'image est homogène par rapport au critère de découpage, sinon le nœud est interne. Il existe plusieurs fonctions permettant d'associer un identificateur à un nœud d'arbre quaternaire [52]. Ces fonctions permettent de retrouver facilement, à partir de l'identificateur de l'image et du nœud d'arbre quaternaire, le quadrant associé dans l'image. Deux nœuds de même identificateur dans deux arbres quaternaires différents sont nommés nœuds homologues. Les nœuds internes d'un arbre quaternaire peuvent contenir de l'information comme par exemple l'histogramme de couleurs ou la signature de la région correspondante.

Certaines approches [55-56] indexent les images par des arbres quaternaires dont le nombre de niveaux est fixe (généralement inférieur à 3). Dans ce cas chaque image est représentée par un arbre quaternaire complet équilibré. Chaque nœud (interne ou feuille) de l'arbre quaternaire contient de l'information sur la région correspondante dans l'image, comme par exemple l'histogramme des caractéristiques visuelles de la région (couleur, texture, forme ou une combinaison des ces caractéristiques). Une telle

structure est appelée histogrammes multi-niveaux(en anglais multi-level histograms), elle permet de filtrer les images au fur et à mesure de la recherche. La figure 2.8 donne un exemple d'histogrammes de couleurs multi-niveaux, chaque nœud de l'arbre quaternaire contenant l'histogramme de couleurs de la région correspondante dans l'image.

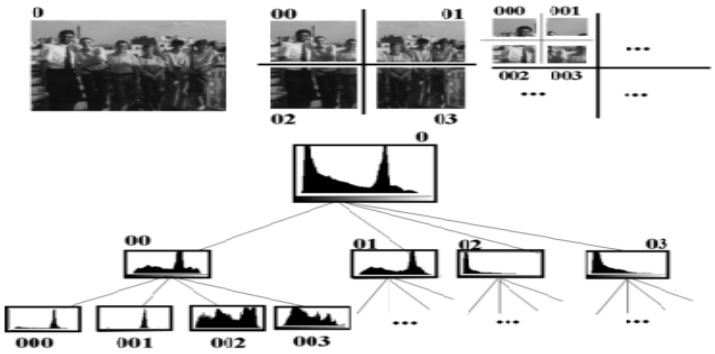


Figure 2.8 : Exemple d'arbre quaternaire d'histogrammes de couleurs multi-niveaux

## 2.3.1 Distances de similarité basées sur les arbres quaternaires

La recherche d'image par le contenu est basée sur la similarité des Caractéristiques visuelles des images. La fonction de distance utilisée pour évaluer la similarité entre images dépend des critères de la recherche, mais également de la représentation des caractéristiques de l'image.

L'idée principale est généralement d'associer à chaque image un arbre quaternaire représentant les caractéristiques de l'image, et de mesurer la similarité des images en utilisant une fonction de distance entre ces arbres [57].

## 2.3.2 Définition générale de la distance

La distance Δ est une distance entre images représentées par des arbres quaternaires [57]. La distance Δ entre-deux images i et j est définie par une somme de distance $\delta_k$ (i , j) entre les nœuds des arbres quaternaires représentant les images i et j , pondérées par des coefficients $C_k$ tel que $C_k$ > 0 :

$$\Delta(i,j) = \frac{\sum_k C_k \delta_k(i,j)}{\sum_k C_k} \qquad (2.1)$$

- Δ (i ,j) est une distance normalisée entre les nœuds homologues k des arbres quaternaires i et j.
- k est l'identificateur d'un nœud pris parmi l'union des identificateurs de nœuds apparaissant dans les arbres quaternaires des images i et j .
- $C_k$ est un coefficient positif représentant le poids du nœud k dans le calcul de la distance.
- Chaque poids $C_k$ est choisi selon l'importance qu'on souhaite donner à certains quadrants d'image par rapport à d'autres dans le calcul de la distance Δ .

### 2.3.3 Cas particuliers de la distance Δ

En fonction des différents poids $C_k$ associés aux nœuds et de la distance choisie entre les nœuds, plusieurs types de distances peuvent être définis à partir de la distance Δ :

- La distance T (T pour Tree) : Cette distance permet d'avoir la comparaison de la structure de deux arbres quaternaires représentant des images, sans tenir compte de la valeur des nœuds feuilles la distance $\delta_k(i,j)$ entre les nœuds d'arbre quaternaire ne prend que 2 valeurs : 0 lorsque les deux nœuds sont tous les deux internes ou tous les deux feuilles et 1 lorsque le nœud est feuille dans un arbre quaternaire et interne dans l'autre ou lorsque le nœud k existe seulement dans un arbre.
- La distance Q (Q pour quadrant) : Cette distance compare deux arbres quaternaires non seulement du point de vue de leur structure, mais également du point de vue des valeurs de leurs nœuds. La distance $\delta_K(i,j)$ entre les nœuds d'arbres quaternaires prend la valeur 0 lorsque tous les nœuds homologues sont tous les deux internes ou tous les deux feuilles avec la même valeur ; la valeur 1 lorsque le nœud est feuille dans un arbre quaternaire et interne dans l'autre ou lorsque le nœud k existe seulement dans un arbre et une valeur comprise entre ]0,1[ lorsque les deux nœuds sont à la même position, mais leurs valeurs sont différentes.
- La distance V (V pour visuel): Lors du calcul de la distance *v* entre deux images i et j, les arbres quaternaires de ces images sont complétés pour avoir la même structure. On ne tient compte alors que des valeurs des nœuds $\delta_k(i,j) = 0$ pour tous les nœuds internes ).

## 2.3.4 L'arbre quaternaire générique

L'arbre quaternaire générique [42] est une structure de données permettant de stocker les images similaires organisées en arbre quaternaire, la similarité des images étant définie par la distance entre les arbres quaternaires les représentant. Cette structure minimise l'espace de stockage par partage des parties communes entre les images, via cette structure un utilisateur peut facilement choisir une ou plusieurs images dans la base de données, un utilisateur peut également modifier une image existante dans la base, insérer ou supprimer des images, extraire des images pour construire des séquences.

### 2.3.4.1 Le concept de partage entre les arbres quaternaires

L'arbre quaternaire générique est basé sur le principe de partage de régions (quadrants) entre images. Soit $I_m$ un ensemble des images. Si un quadrant *q* a la même valeur dans un ensemble $I'_m \subset I_m$ cette valeur n'est stockée qu'une seule fois dans la base et est associée à l'ensemble des identificateurs des images $I'_m$. Dans ce cas on parle de partage explicite, parce que l'identificateur de chaque image partageant cette valeur apparait explicitement dans la liste des images associées à cette valeur.

Si les images de l'ensemble $I_m$ sont organisées en arborescence, chaque image excepté la racine de l'arbre, à une mère unique et un nombre indéfini d'images filles. Par conséquent la règle de partage implicite suivante peut être introduite : *'' excepté lorsque l'identificateur d'une image i est implicitement associé avec une autre valeur v, l'image i partage implicitement la valeur associée à son image mère ''*.

Si l'arbre organisant les images est stocké, cette règle de partage implicite permet d'avoir une représentation compacte de l'ensemble d'images, en particulier lorsqu'un grand nombre d'images partagent des valeurs de quadrants dans plusieurs branches de l'arbre.

### 2.3.4.2 Similarité entre images

Les images sont regroupées, dans la base de données ; en fonction d'une distance de similarité entre les arbres quaternaires qui les représentent. Cette distance, appelée *Q-similarité* est proposée afin d'optimiser le stockage des images dans la base. La distance de *Q-similarité* entre deux images est définie par le nombre des nœuds différents (de même identificateur et de valeur différente) dans les deux arbres quaternaires représentants les images, divisé par le nombre d'identificateurs de nœud (sans doublon) apparaissant dans l'union des nœuds des arbres quaternaires des

images. De manière plus formelle, on note *S (i, i')* l'ensemble des nœuds différents entre les arbres quaternaires des images *i*, *i'*. On note *U (i, i')* l'ensemble (sans doublon) des identificateurs de nœud apparaissant dans les arbres quaternaires des images *i* et *i'*. On note $|S(i,i')|$ (resp. $|U(i,i')|$) le nombre d'éléments de *S (i, i')* (resp. *U (i, i')*) .la distance de *Q-similarité* des images *i* et *i'* notée *d (i, i')*, est calculée par l'équation suivante :

$$d(i,i') = \frac{|S(i,i')|}{|U(i,i')|} = \frac{nombre\ de\ nœuds\ différents}{Total\ des\ identificateurs\ de\ nœud\ (sans\ doublon)} \qquad (2.2)$$

### 2.3.4.3 L'arbre d'images

En conséquence de la règle de partage implicite, les images représentées par un arbre quaternaire générique sont organisées à l'aide d'une structure arborescente particulière, l'arbre d'images. Lorsqu'une nouvelle image est insérée dans l'arbre d'images, elle est insérée comme fille de l'image dont elle est la plus similaire, c'est-à-dire dont la distance entre l'arbre quaternaire associée et celui de l'image à insérer est la plus proche de 0.

### 2.3.4.4 Nœuds génériques

La représentation et le stockage d'un ensemble des images similaire sont effectués dans un arbre quaternaire générique, dont les nœuds sont appelés nœuds générique. Pour chaque nœud apparaissant dans l'arbre quaternaire d'une image, il existe un nœud générique ayant le même identificateur dans l'arbre quaternaire générique. Un nœud générique *n* représente tous les nœuds *n* des arbres quaternaires des images de la base. Il contient toute l'information nécessaire pour recomposer la valeur du nœud de même identification dans chaque arbre quaternaire.

La valeur d'un nœud est soit $\perp$ qui signifie que le nœud n'existe pas, soit *int* qui signifie qu'il est interne (il y a quatre fils). Chaque nœud générique peut être vu comme un tableau ayant plusieurs lignes. Chaque ligne *l* de nœud générique *n* contient une liste d'identificateurs d'images et une valeur *v* de nœud d'arbre quaternaire. La valeur *v* signifie que tous les nœuds identifiés par *n* ont pour valeur *v* dans les arbres quaternaires des images dont l'identificateur *i* apparait dans la liste.

De plus par application de la règle de partage implicite, on déduit que les nœuds *n* des arbres quaternaires de toutes les images descendantes des images de la liste, dans

l'arbre d'images partagent implicitement cette valeur, excepté lorsqu'un identificateur d'image descendante apparait explicitement dans une autre ligne du nœud générique. La figure 2.9 montre une présentation de quatre images présentées (a,b,c,d) en arbre quaternaire générique.

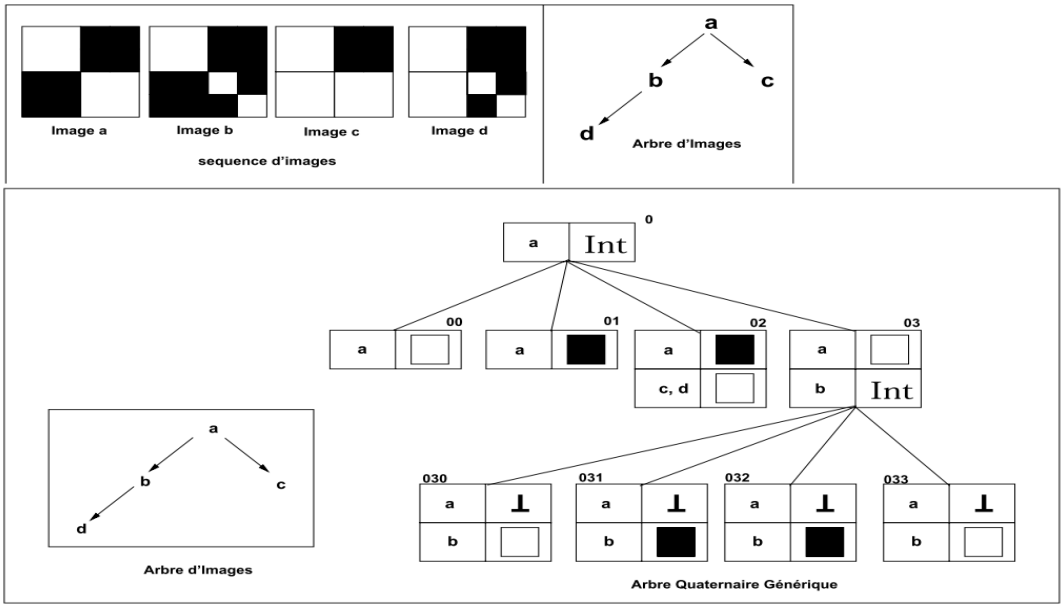


Figure 2.9 : Exemple d'arbre quaternaire générique

## 2.3.5 L'arbre R_générique

L'arbre R_Générique permet de gérer et de stocker des images similaires organisées en arbre R, la similarité des images étant définie par la distance entre les arbres qui les représentent, l'arbre R_Générique minimise l'espace de stockage par partage des parties communes entre les images [37]. Cette structure est basée sur le principe des arbres quaternaires générique [42].

## 2.4 Problème de La malédiction de dimension

Des phénomènes mathématiques particulièrement peuvent être observés quand la dimension de l'espace des données augmente. Ces effets portent le nom de malédiction de la dimension, qui est une traduction de l'anglais curse of dimensionality. Ils sont notamment passés en revue dans les études [58],[59] menées pour l'indexation multidimensionnelle, mais seulement dans le cadre d'une distribution uniforme des données. De nombreuses techniques d'indexation sont basées sur le partitionnement de

l'espace en volumes englobant les données. Or on peut observer que les volumes et les surfaces augmentent de manière exponentielle avec la dimension de l'espace, ce qui conduit à une forte augmentation du temps de réponse des algorithmes de recherche. Au-delà d'une certaine dimension, un parcours séquentiel de l'espace devient même plus performant qu'un parcours d'index. Les problèmes qui se posent dans les espaces de grandes dimensions sont nombreux et il est généralement difficile d'étendre dans ces espaces les techniques que l'on a dans les espaces à deux ou à trois dimensions.

L'un des premiers problèmes qui rendent les techniques d'indexation fondées sur le partitionnement de l'espace inefficace est celui de la croissance exponentielle du nombre de cellules en fonction de la dimension. Cela pose de sérieux problèmes pour les procédures de recherche. En effet, dans le cas où un vecteur requête se trouve suffisamment près d'une frontière entre cellules, le processus de recherche des plus proches voisins doit nécessairement examiner la ou les cellules voisines ce qui augmente par conséquent le temps de la recherche. Les techniques d'indexation basées sur le partitionnement de données regroupent les données en des formes géométriques particulières (rectangles, sphères…). Toutefois les propriétés géométriques de ces formes ne sont plus maitrisables en grande dimension. Par exemple, il est simple de montrer que la limite du volume d'une hyper-sphère lorsque le nombre de dimensions tend vers l'infini est nulle. Si les données ont une répartition homogène dans l'espace, l'observation précédente implique que la grande majorité des données se situent à l'extérieur de la sphère. Ce phénomène que nous retrouverons dans les volumes de recouvrement des méthodes telles que la famille du R-Tree, SS-Tree, SR-Tree est appelé phénomène de l'espace vide, il complique le groupement des vecteurs en paquets, car le nombre de vecteurs n'est pas suffisamment élevé par rapport au nombre de dimension. Ceci produit par conséquent des formes géométriques volumineuses qui se chevauchent et avec peu de vecteurs diminuant ainsi la performance des index en grande dimension [18].

Ces problèmes peuvent être résumés comme suit :

- ✓ La distance d'un point donné au point le plus proche tend vers la distance au point le plus éloigné.
- ✓ Faible occupation de l'espace.
- ✓ En moyenne toutes les feuilles des index sont visitées.

## 2.5 Conclusion

Les techniques d'indexation restent très souvent mieux adaptées à des vecteurs de petites dimensions ; or dans des applications réelles, les données sont généralement représentées par un grand nombre de caractéristiques ; ce qui rend l'utilisation de la plupart des index multidimensionnels difficile entrainant ainsi une diminution de leur performance qui est proportionnelle au nombre de dimensions. Pour pallier au problème de l'espace de grande dimension, une réduction de dimension s'avère nécessaire. Plusieurs techniques en statistique et analyse de données permettent de réduire un espace original en un autre espace en gardant le maximum d'information portée par les données dans leurs espaces originaux [25].

L'arbre quaternaire est très utilisé dans le domaine des images, aussi bien pour le stockage, la compression, l'extraction de l'information, mais aussi pour la recherche d'image par le contenu. Cette forme de structure est très utilisée pour stocker les différentes régions de l'image et de filtrer les images en augmentant au fur et à mesure le niveau de détail. Ceci va permettre de faire la comparaison globale entre deux images, d'effectuer des requêtes sur des régions d'images telles que ''*trouver toutes les images de la base ayant la région nord-ouest et la région sud-est similaires à celle de l'image requête*''.

# CHAPITRE 3
# SEGMENTATION EN PLAN DE VIDEOS BASEE SUR LA METHODE D'ARBRE QUATERNAIRE ET LES DESCRIPTEURS VISUELS

## 3.1 Introduction

Suite aux notions de base présentées précédemment, ce chapitre s'intéresse à décrire les différentes étapes de notre technique de segmentation en plans de vidéo basée sur la méthode d'arbre quaternaire et les descripteurs visuels. Nous présentons en premier lieu les descripteurs visuels utilisés. Par la suite, nous décrivons la sauvegarde des descripteurs visuels dans les différents niveaux de l'arbre quaternaire et enfin nous abordons les étapes nécessaires pour la segmentation en plans de la vidéo dont l'algorithme est basé sur la manipulation de la structure d'arbre quaternaire.

## 3.2 Les descripteurs utilisés

La comparaison directe des images entre elles n'est pas envisageable. il est donc nécessaire d'extraire des informations représentatives. Afin de caractériser l'image, nous avons utilisé des descripteurs exploitant la forme, la couleur et la texture de l'image que nous allons détailler ci-dessous.

### 3.2.1 Descripteur de la forme

La forme est l'un des attributs bas niveau le plus utilisé pour décrire le contenu visuel des images. Les descripteurs de formes sont utilisés pour décrire la structure géométrique générique du contenu visuel. Les descripteurs de forme peuvent être classés en deux familles :

- Descripteurs orientés région : qui décrivent les objets selon la distribution spatiale des pixels qui les constituent.

- Descripteurs orientés contour : qui décrivent les objets selon leur contour externe.

Dans le cadre de ce travail, nous considérons la forme d'une image comme étant celle formée uniquement de ses contours. Le contour d'un objet n'est pas toujours facile à extraire ou à détecter surtout quand l'image est bruitée. Dans ce cas, on lui fait subir un filtrage. Cette opération de prétraitement dépend du domaine de l'application. Si l'objet d'intérêt est connu d'avance, par exemple il est plus foncé que le fond, alors un simple seuillage d'intensité peut isoler le bruit. Pour des scènes plus complexes, les transformations invariantes au changement d'échelle, à la translation et à la rotation peuvent être nécessaires. Une fois l'objet détecté et localisé, sa forme peut être trouvée par un des algorithmes de détection.

Avant de procéder à la description du système de détection de contours, nous allons commencer par donner quelques notions fondamentales nécessaires au développement de ce système.

#### 3.2.1.1. Filtrage d'images

Le filtrage est une opération qui a pour but d'extraire une information ou d'améliorer l'aspect de l'image, par exemple en éliminant un bruit (lignage, speckle des images radar, etc.) ou en améliorant les contours d'une image floue. Nous distinguons deux types de filtrage :

- ✓ Le filtrage global : chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ.
- ✓ Le filtrage local **:** chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire.

Dans le cadre de ce travail, nous avons opté pour un filtrage local linéaire. Ce dernier transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique nommée convolution [60]. Il existe plusieurs filtres linéaires, mais ici nous nous intéressons uniquement au filtre gaussien qui a fait l'objet de notre étude. Le filtre gaussien, appelé également *gaussian filtering* est obtenu par convolution de l'image avec une gaussienne [60]. Nous rappelons l'expression d'une gaussienne en dimension 2, de moyenne nulle :

$$G_\sigma(X) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{|X|}{2\sigma^2}\right)} \qquad (3.1)$$

Où la largeur du filtre est donnée par son écart-type σ. Le calcul de la largeur du filtre de part et d'autre du point central correspond à Ent+(3σ) sachant que Ent+(.) est l'entier supérieur. Par ailleurs, la largeur totale du filtre est donnée par : 2Ent+(3σ)+1.

### 3.2.1.2 Détection des contours de l'image

La notion de contour est reliée à celle de variation en chaque pixel. Une variation existe si le gradient est localement maximum ou si la dérivée seconde présente un passage par zéro. Le gradient repose sur la définition de deux masques H1 et H2 qui calculent le gradient de l'image dans deux directions orthogonales. Le descripteur de contours repose sur les étapes suivantes :

a. Initialiser le filtre gaussien.
b. Convertir l'image en niveaux de gris.
c. Calculer le filtre en utilisant l'équation de la gaussienne.
d. Appliquer la convolution avec le filtre gaussien sur l'image en niveaux de gris.
e. Détecter des contours à partir de l'image convoluée.
f. Calculer le gradient par le noyau Sobel (direction horizontale).
g. Calculer le gradient par le noyau Sobel (direction verticale).
h. Générer le vecteur d'orientation de contours.

La détection des contours est faite par l'approche de convolution en utilisant le filtre de Sobel [61]. La plupart des procédés de détection de contour travaillent sur l'hypothèse que le contour se produit lorsqu'il y a une discontinuité de la fonction d'intensité ou un gradient d'intensité. L'utilisation de cette hypothèse, si l'on prend la dérivée de la valeur d'intensité à travers l'image est de trouver les points où la dérivée est optimale. La méthode du gradient détecte les contours en recherchant le maximum et le minimum dans la première dérivée de l'image. Le détecteur de contours de Sobel est un exemple de la méthode du gradient.L'opérateur de Sobel est un opérateur discret de différenciation, il calcule une approximation du gradient de l'intensité de l'image [61- 62]. L'opérateur de Sobel permet d'estimer localement la norme du gradient

spatial bidimensionnel d'une image en niveau de gris. Il amplifie les régions de fortes variations locales d'intensité correspondant aux contours. Le calcul de gradient est mené par l'intermédiaire de deux masques H1 et H2, donnés ci-dessous pour les contours horizontaux puis verticaux.

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

H1

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

H2

Un exemple de cet opérateur est donné dans la figure ci-dessous :

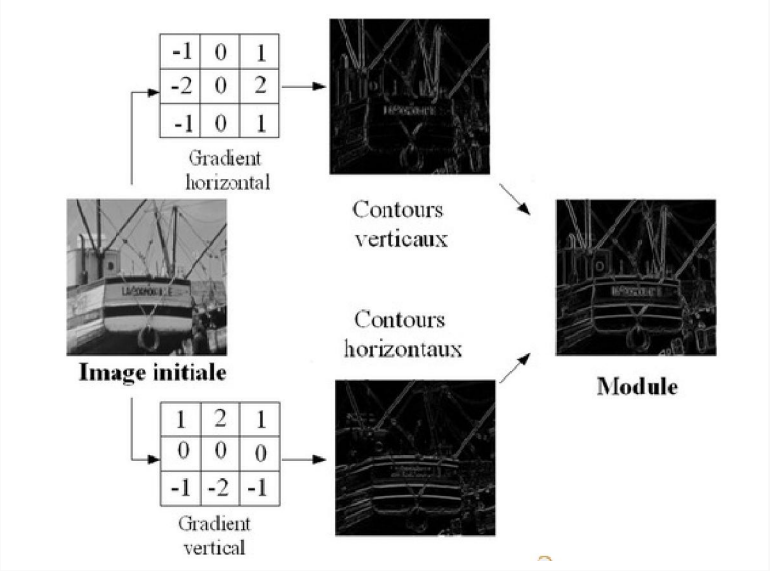


Figure 3.1 Exemple d'application du filtre de Sobel

## 3.2.2 Descripteur de la texture

La texture de l'image représente une primitive visuelle de première importance pour la navigation et la recherche par similarité dans de larges collections d'images. Le descripteur Histogramme de Contours (Edge Histogram Descriptor) est un descripteur de texture retenu dans la norme MPEG7 [63]. Il représente la distribution spatiale de 5 types de contours pour 16 régions locales d'une image donnée. Il s'agit de quatre contours directionnels (horizontal (0°), vertical (90°), deux diagonaux (45° et 135°)) et un contour non directionnel (orientation non spécifique) pour chaque région de l'image. Pour leur détection, on utilise un banc de cinq filtres linéaires adaptés à chacune de ces classes (Figure 3.2). La distribution de contours ne constitue pas seulement une bonne

représentation de texture, mais elle est aussi utile pour la comparaison des images en l'absence de toute texture homogène [41][64].

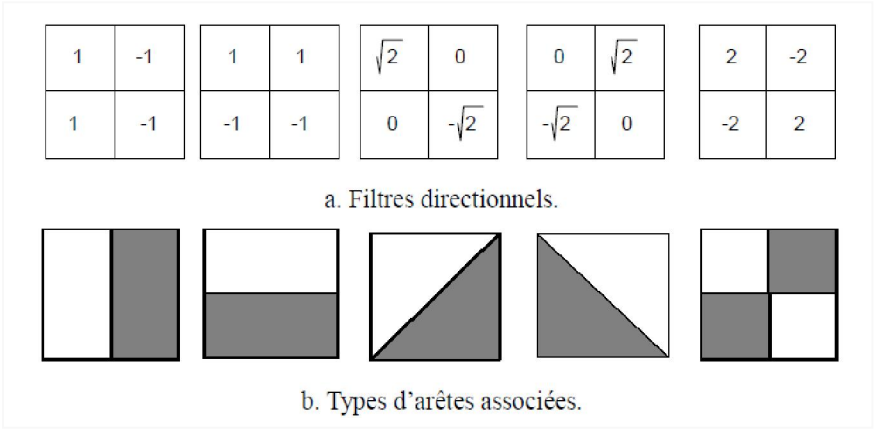


Figure 3.2 Détection des orientations de contour par filtres directionnels adaptés

Les différentes étapes d'extraction du descripteur EHD peuvent être résumées comme suit [41] :

a. Partitionnement de l'image en 4x4 sous images (16 sous images).
b. Division de chaque sous images en 16 images-blocs.
c. Division de chaque bloc en 2x2 macro-blocs.
d. Calcul de l'intensité de couleur pour chaque macro-bloc.
e. Application des opérateurs de détection de contours.
f. Génération du vecteur descripteur de l'image associée à l'histogramme local de contours.

## 3.2.3 Descripteur de couleur

Attribut largement utilisé dans le domaine de l'indexation d'images et la recherche par le contenu, la couleur a été également considérée dans le cadre de notre travail. Le descripteur '*histogramme scalable de couleur (SCD)*', choisi dans notre étude, propose une représentation multi-résolution, par transformée de Haar [65] d'un histogramme de couleur spécifique dans l'espace HSV( Hue, Saturation, Value).

Le descripteur correspondant à l'histogramme scalable de couleur est défini comme une quantification uniforme de l'espace de couleur Hue-Saturation-Value (HSV). Les valeurs des bins subissent une quantification non-linéaire pour réaliser un encodage efficace. La différence réside dans l'utilisation de la transformation de Haar (filtre de Haar) par le

SCD. Cette transformation fournit une description compacte et une représentation multi-échelles de l'histogramme. L'intérêt de cette multi-résolutions est le passage du grossier au raffinement. Ce descripteur ne tient pas compte de la structure locale de la couleur. Le codage par la transformation de Haar est utilisé pour réduire le nombre de bins dans l'histogramme original à 16, 32, 64, 128 ou 256 bins [63].

Les Étapes d'extraction du descripteur SCD sont représentées ci-dessous :

a. Construire l'histogramme de couleur d'une image donnée dans l'espace HSV (Hue, Saturation, Value).
b. Exécuter la procédure de quantification sur l'histogramme obtenu en (a).
c. Appliquer la transformation de Haar suivi d'une quantification linéaire.
d. Générer le descripteur SCD (Scalable Color Descriptor).

## 3.3 La technique adoptée pour la segmentation en plan de la vidéo

L'implémentation de l'approche proposée est divisée en trois étapes dont chacune est caractérisée par des actions dont le but est de concevoir et de réaliser un système de segmentation en plans de la vidéo. Les étapes de notre démarche sont les suivantes :

*Étape 1:* dans cette étape, nous découpons la vidéo en une série d'images individuelles. Ceci se fait à l'aide de logiciels de traitement de vidéo existant sur le marché comme Aoao photo digital studio vidéo to picture 3.7. Les images obtenues nous permettront par la suite de définir des plans d'images en utilisant des distances de similarité. Cette étape est décrié par la figure 3.3. .


Figure 3.3 : Découpage vidéo en image individuelle

*Étape 2*:

En premier lieu cette étape sert à construire l'arbre quaternaire équilibré à trois niveaux pour chaque image obtenue dans l'étape 1. Pour cela chaque image est découpée spatialement en 16 quadrants pour être présentée par un descripteur multi-niveau. La figure 3.4 présente la description cette procédure de découpage.

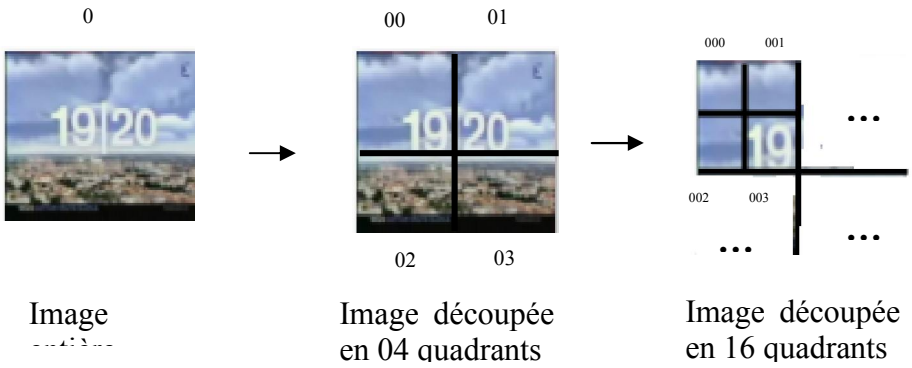


Figure 3.4: Découpage d'une image en 4 puis en 16 quadrands

Par la suite, chaque image découpée en 16 quadrants sera représentée par un arbre quaternaire équilibrée à trois niveaux dont le nombre de nœuds est égal à 16. Chaque nœud est identifié par un numéro. La figure 3.5 montre la construction de l'arbre quaternaire pour chaque image individuelle.

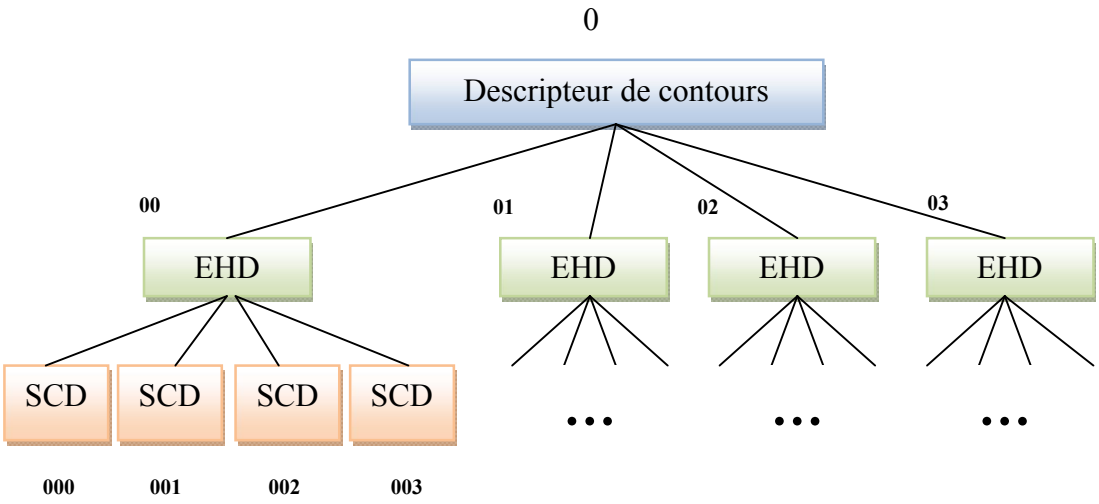


Figure 3.5: Construction de l'arbre quaternaire pour chaque image individuelle

EHD : *Histogramme d'orientation des contours*

SCD : *l'histogramme scalable de couleur*

*Étape 3*:

Pour exploiter l'approche de l'arbre quaternaire, dans la segmentation en plan de la vidéo nous proposons un algorithme qui tient compte la localisation spatiale des images. Dans notre algorithme pour chaque niveau d'arbre, nous avons sauvegardé un descripteur visuel différent du niveau inférieur.

*Premièrement* : on a sauvegardé dans le nœud d'identifiant 0 qui correspond au premier niveau de l'arbre quaternaire le descripteur qui fait la détection des contours en se basant sur l'approche de convolution en utilisant le filtre de Sobel [61] .

*Deuxièmement :* on a stocké dans chaque nœud d'identifiants (00,01,02,03) qui correspond au deuxième niveau de l'arbre quaternaire le descripteur de texture (EHD) retenu dans la norme MPEG7.

*Troisièmement :* pour chaque nœud d'identifiant *{ 00i, 01i , 02i , 03i tel que i=0 à 3 }* qui correspond au troisième niveau, le descripteur l'histogramme scalable de couleur (SCD) de la norme MPEG7 est sauvegardé dans les 16 nœuds. Notre algorithme permet de calculer la distance inter-quadrants par niveau. Ce qui rend l'utilisation de trois distances présentées comme suit :

- ✓ *D0* = est la distance euclidienne entre les deux racines associées aux deux arbres quaternaires. Ce qui traduit par la distance euclidienne entre les deux descripteurs du contour sauvegardés dans les quadrants d'identificateurs *0* avec un poids de $C_k = 1$ .
- ✓ *D1=(D00+D01+D02+D03)*1/4* , tel que *D0i* avec *( i=0 à 3)* est la distance euclidienne entre les deux quadrants d'identificateurs *0i* avec *(i=0 à 3)* associées aux deux arbres quaternaires.
- ✓ *D2=D00i+D01i+D02i+D03i *(1/16)* tel que *D00i* ,*D01i, D02i* et *D03i* avec *( i=0 à 3)* est la distance euclidienne entre les deux quadrants respectivement d'identificateurs *00i ,01i, 02i* et *03i* ,avec *(i=0 à 3)* associées aux deux arbres quaternaires.

*D =(D0+ D1+D2)/(1+1/4+1/16)* est la distance finale entre les deux arbres quaternaires. Cette distance repose sur la distance présentée par l'équation (2.1) :

Les coefficients $C_k$ Associés aux nœuds sont choisis de sorte qu'ils soient proportionnels à la surface des quadrants correspondant dans l'image. Ce qui traduit par la formule suivante :

$$C_k = 4^{-p} \qquad (3.2)$$

Plus le nœud est situé profondément dans l'arbre plus la valeur de son coefficient est faible , donc son poids dans le calcul de la distance Δ est faible. P désigne le niveau de l'arbre quaternaire.( pour la racine P=0,pour le niveau 1 P=1 et p=2 pour le niveau 2). En ce qui suit, nous présenterons notre propre algorithme de segmentation en plan de la vidéo permettant la manipulation de l'arbre quaternaire [66]:

### 3.3.1 L'algorithme de détection de changement des plans

*Début*

*1 : Construire l'ensemble E des images individuelles à partir de la vidéo en utilisant des algorithmes de découpage.*

*2 : Construction de l'arbre quaternaire à trois niveaux pour chaque image obtenue à l'étape 1 suivant les instructions suivantes :*

- ✓ *Sauvegarder dans le niveau 0 le descripteur du contour*
- ✓ *Sauvegarder dans le niveau 1 le descripteur de la texture*
- ✓ *Sauvegarder dans le niveau 2 le descripteur de la couleur*

*Le 1$^{er}$ plan de la vidéo contient uniquement la 1$^{ére}$ image de l'ensemble E.*

*3 : Calcul de la distance de Q similarité entre l'image suivante et la 1$^{ére}$ image du plan courant selon la procédure suivante :*

- *Pour le niveau 0 : calculer la distance entre la racine de l'arbre quaternaire de l'image suivante et la première image du plan courant. La distance $\delta_k(i,j)$ =distance euclidienne entre les descripteurs du contour. Le poids $C_k$=1 (pour la racine p=0).*
- *Pour le niveau 1 : pour chaque quadrant (en totalité 4 nœuds) il y'aura un calcul de distance $\delta_k(i,j)$ = distance euclidienne entre les descripteurs de la texture. Nous faisons la somme de ces distances puis nous les multiplions par le poids $C_k$ =1/4 (pour le niveau 1 p=1).*
- *Pour le niveau 2 : pour chaque quadrant (en totalité 16 nœuds), on calcule les distances inter-quadrants du niveau 2 en utilisant la même démarche déjà citée aux niveaux 0,et 1 sachant que le descripteur est la couleur et $C_k$=1/16 (pour le niveau 2 p=2).*
- *À chaque niveau, une distance est calculée. La moyenne de toutes les distances permet d'obtenir, la distance finale entre l'image suivante et la première image du plan courant. Le calcul de la distance entre deux arbres quaternaires repose sur la distance Δ .*

*4 : Si la distance est inférieure au seuil (fixé de manière empirique) alors cette image fait partie de ce plan .*

*4.1. Tant qu'on n'a pas atteint la fin de l'ensemble E, on passe à l'image suivante de la séquence.*

*4.2. Revenir a l'étape 4.*

*5 : Sinon l'image courante génère un nouveau plan et aller à l'étape 3.*

*6 : Dès qu'on atteint la fin de l'ensemble E, on s'arrête.*

*Fin.*

## 3.4 Complexité algorithmique

La complexité d'un algorithme permet de calculer la mesure de l'efficacité d'un algorithme pour un type de ressource. Un algorithme a un coût qui est lié au nombre d'opérations effectuées et à l'espace mémoire occupé par les données. Cela revient à calculer la complexité d'un algorithme en temps d'exécution ce qu'on appelle complexité temporelle ou bien en espace mémoire. Dans notre cas nous nous sommes intéressés à la complexité en temps d'exécution.

Notre algorithme est divisé en six étapes, ce qui nous permet d'associer pour chaque étape une complexité algorithmique $O_i$ , tel que : *i= 1* à *6*.

### 3.4.1 Calcul de la complexité

*Etape 1 :* le découpage de la vidéo en *N* images est une boucle, ce qui se traduit par une complexité $O_1(N)$

*Etape 2 :* pour chaque image il y aura 21 quadrants donc il faudra calculer la complexité pour les trois Descripteurs (la forme , la texture et la couleur ) qui sont implémenté en boucle, donc on aura une complexité égale à $T_1(N)+4T_2(N)+16T_3(N)=T(N)$, Quelque soit la taille de l'image , il y aura 21 nœuds ,alors la construction de l'arbre est $T_4(1)$, Construire l'arbre pour chaque image et en calculant leur distance, ce qui ramène à définir la complexité suivante :

$$\sum_{i=1}^{N}\big(T(N)+T_4(1)\big)=N.Max\big(T(N),T_4(1)\big)=O_2(N)$$

N est le nombre d'image de la séquence vidéo

*Etape 3 :* calcul de la distance, donc c'est $O_3(1)$.

*Etape 4 :* pour les trois niveaux nous avons la complexité suivante :

Niveau 0 : $1.T_0(N)$

Niveau 1 : $4.T_1(N)$

Niveau 2 : $16.T_2(N)$

Pour le calcul de Q similarité :$T_4(N)= 1.T_0(N)+ 4.T_1(N)+ 16.T_2(N)$

Ceci se répète *(N-1)* fois pour chaque images, donc la complexité de cette étape est :

(N-1)$O_4(N)=O_4(N)$ (N : est le nombre d'images de la séquence vidéo)

*Etape 5 :* la complexité égale à $O_5(1)$ Car il y a un traitement indépendant du nombre d'image de la vidéo plus une affectation.

*Etape 6 :* il y a une affectation et une boucle, donc c'est $T_1(1)+T_2(N)=O_6(N)$

La complexité finale de notre algorithme est :
T(N) =Max($O_1(N)$, $O_2(N)$, $O_3(1)$, $O_4(N)O_5(1)$, $O_6(N)$)=$O(N)$

## 3.4.2 discussion

La complexité de notre algorithme est linéaire $O(N)$. Ceci nous permet d'éviter la complexité exponentielle, qui caractérise les algorithmes intrinsèquement complexes. Dans notre cas si nous avons un processeur pouvant effectuer $10^9$ opérations par seconde, nous obtenons les résultats suivants :

- ✓ pour N= $10^1$ alors $O(N)$ tend vers 0.
- ✓ pour N=$10^3$ alors $O(N)$ tend vers 0.
- ✓ pour N= $10^6$ alors $O(N)$ tend vers 0.
- ✓ pour N=$10^9$ alors $O(N)$ tend vers 1 seconde.
- ✓ pour N=$10^{12}$ alors $O(N)$ tend vers ∞.

Nous remarquons que dans certains cas, l'algorithme est réalisable dans un temps quasi-immédiat (inferieur à une seconde). Dans d'autres cas, il peut y avoir un temps considérable pour son exécution.

## 3.5 Conclusion

Dans ce chapitre nous avons présenté la description générale de chaque descripteur utilisé dans la technique que nous avons proposé pour la segmentation en plans de la vidéo. Le présent système se base sur la structure d'arbre quaternaire équilibré à trois niveaux. Chaque niveau de un à trois sauvegarde respectivement les descripteurs visuels de la forme, la texture et la couleur (respectivement EHD et SCD de la norme MPEG7).
L'algorithme présenté permet la manipulation de la structure d'arbre quaternaire spécifié et génère les plans de la vidéo à partir d'un ensemble d'images obtenu par le découpage de vidéo. La similarité entre les images d'un plan vidéo est obtenue par le calcul de Q-similarité entre deux arbres quaternaires. Dans ce qui suit, nous montrons les résultats de la segmentation en plans de la vidéo obtenue pour différentes vidéos afin de valider notre approche.

# CHAPITRE 4
# EXPERIMENTATIONS ET DISCUSSIONS

## 4.1 Introduction

L'objectif de ce chapitre est de présenter une vue globale sur la mise en place de notre technique de segmentation en plans de la vidéo. Il illustre les résultats d'expérimentation par l'algorithme présenté dans le chapitre précédent. L'algorithme de détection de changement de plans admet en entrée une séquence d'images obtenue par le découpage d'une vidéo pour produire un ensemble de plans vidéo. Nous abordons plus précisément l'effet du seuil sur la détection du changement de plans brusque et progressif pour plusieurs séquences d'images ; chaque image étant représentée par une structure d'arbre quaternaire. L'expérimentation est répétée pour plusieurs valeurs du seuil en considérant des vidéos à différents contenus.

## 4.2 Implémentation

Notons que notre technique de segmentation en plan de la vidéo basée sur l'arbre quaternaire et les descripteurs visuels a été implémentée sous l'environnement Microsoft Visual Studio 2008 sur un micro ordinateur ayant une fréquence de 2,53 GHZ, mémoire vive de 2 Go et disque dur de 230 Go.
La technique proposée est divisée en trois étapes :

*Étape 1:* dans cette étape nous découpons une vidéo en une série d'images individuelles. Ceci se fait à l'aide de logiciels de traitement de vidéo existant sur le marché. Dans notre cas nous avons utilisé le logiciel Aoao photo digital studio vidéo to picture 3.7 [67]. Ces images nous permettront par la suite de définir des plans d'images en utilisant des distances de similarité figure 4.1.

*Étape 2*: Cette étape sert à construire l'arbre quaternaire équilibré à trois niveaux pour chaque image obtenue dans l'étape 1 ; ceci nécessite de calculer les distances de Q-Similarité entre les images.


Figure 4.1 : Le découpage en images individuelles de la vidéo documentaire par Aoao photo digital studio vidéo to picture 3.7

*Étape 3*: dans cette étape nous parcourons l'ensemble des images et on calcule la distance de similarité entre chaque deux image consécutive. La distance trouvée est comparé au seuil qui a été fixé au début de l'algorithme. Si la distance est supérieure au seuil nous concluons que l'image suivante forme un nouveau plan.

## 4.3 Mesure d'évaluation de la méthode de segmentation

Pour l'évaluation de notre technique de détection de changement de plans, plusieurs critères sont utilisés [41] [68]. Les plus adoptés sont :

Le ''Rappel' et la ''Précision''. Dans notre cas, l'évaluation consiste à calculer le nombre de plans détectés, le nombre de plan correctement détectés et le nombre de plan en référence. Nous avons utilisé le logiciel Movie Maker pour détecter manuellement le nombre des plans en référence [41].

Le rappel est le rapport entre le nombre des plans correctement détectées et le nombre des plans en référence.

$$\text{Rappel} = \frac{\text{plans correctement détectés}}{\text{plans en référence}} \qquad (4.1)$$

La précision est le rapport entre le nombre des plans correctement détectés et le nombre des plans détectés.

$$\text{Précision} = \frac{\text{plans correctement détectés}}{\text{plans détectés}} \qquad (4.2)$$

- ✓ Les plans détectés sont des plans obtenus par notre algorithme.
- ✓ Les plans correctement détectés sont des plans corrects (par rapport à des plans en référence) détectés par notre algorithme.

## 4.4 Présentation des vidéos de test

Nous avons travaillé principalement sur six vidéos dans le but d'évaluer et de valider notre système de segmentation en plan de vidéo. Les vidéos utilisées sont de types différents (journal télévisé, film, dessins animés, football, documentaire,

publicité) avec une variété de contenu (effets spéciaux, sous-titrages, etc.). Ces vidéos sont de type AVI et de fréquence d'image égale à 25 images par seconde. Les séquences vidéo utilisées sont données par la figure 4.2 :


Figure 4.2 : Liste des vidéos

Le tableau suivant résume les caractéristiques de six vidéos :

| Type de la vidéo | Format de fichier | Durée en seconde | Nombre de frame /seconde | Nombre d'images | Nombre de plans en référence par Movie Macker |
|---|---|---|---|---|---|
| **film** | AVI | 9 | 15 | 132 | 10 |
| **Dessin animée** | AVI | 9 | 15 | 140 | 7 |
| **Football** | AVI | 11 | 15 | 181 | 4 |
| **documentaire** | AVI | 9 | 15 | 151 | 7 |
| **Publicité** | AVI | 10 | 15 | 163 | 4 |
| **Journal télévisé** | AVI | 12 | 15 | 197 | 5 |

Tableau 4.1 les caractéristiques des videos

## 4.5 Résultats et discussions :

Nous présentons dans cette partie quelques exemples de résultats expérimentaux obtenus par la technique proposée. Ces résultats ont été obtenus en appliquant différents seuils correspondant à 25, 27, 29, 30, 31 et 33.

- ✓ Seuil =25

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 34,78 % | 72,72 % |
| dessins animés | 46,15 % | 85,71 % |
| Football | 66,67 % | 66,67 % |
| Documentaire | 100% | 100 % |
| Publicité | 50 % | 66,67 % |
| Journal télévisé | 60 % | 75 % |

Tableau 4.2 : Résultats obtenus en appliquant un seuil à 25

A partir de ce tableau, nous remarquons que les résultats sont intéressants pour la vidéo documentaire (un rappel=100% et une précision=100%). Cela est dû à la structure de l'arbre quaternaire qui permet de stocker les caractéristiques des différentes régions d'images et de filtrer les images en augmentant au fur et à mesure le niveau des détails d'une part. Par ailleurs, la valeur du seuil=25 qui représente la distance entre deux images successives représentées par les arbres quaternaires est la plus petite valeur par rapport aux autres valeurs du seuil dans notre partie expérimentale. Notons que cette petite valeur affecte grandement le résultat lorsqu'il y a un minimum de changement du contenu visuel entre deux images successives dans notre technique. Cette valeur est calculée à partir d'une combinaison de trois descripteurs (contour, texture et couleur).

L'utilisation des descripteurs correspondant aux contour, EHD (texture) et SCD (couleur) respectivement dans les niveaux de l'arbre quaternaire 1,2 et 3 joue un rôle important. En effet, le descripteur SCD a été stocké dans le troisième niveau de l'arbre. Ce descripteur offre de bonnes performances lorsque les couleurs sont quasi-homogènes, comme dans le cas de la vidéo documentaire et film.

Le descripteur d'orientation de contour (EHD) stocké au deuxième niveau de l'arbre est invariant aux rotations. De plus, ce descripteur donne des résultats intéressants lorsque les images présentent des textures uniformes, comme dans le cas des vidéos dessins animés, film et documentaire. Les deux derniers descripteurs EHD et SCD sont combinés avec le descripteur de contour, stocké dans le premier niveau de l'arbre quaternaire, permettent de tenir compte de la localisation des caractéristiques dans le calcul de similarité des images, ce qui influe sur les résultats présentés dans le tableau 4.2.

Nous pouvons constater que la continuité fine du contenu visuel des images successives qui compose la vidéo documentaire permet de détecter facilement le passage d'un plan à un autre sans image de transition, c'est le cas des plans brusques. Pour les cinq vidéos restantes, les résultats ne sont pas satisfaisants. Ceci se traduit par la même raison citée auparavant. En outre, les caractéristiques du contenu visuel des images qui compose ces vidéos influent grandement sur les résultats. Nous constatons que la vidéo *film* présente le moins bon résultat. C'est en raison du au contenu des plans de cette vidéo qui correspond à une séquence d'images discontinue : il y a une rupture nette entre l'image et celle qui suit (aucune ressemblance). Pour la vidéo *dessins anim*és et *journal télévisé*, les résultats de la précision sont faibles car l'image disparait progressivement lors d'une transition vers un autre état ; c'est le cas des plans progressifs (fondus enchainés).
Nous avons à nouveau mené les mêmes expérimentations en considérant cette fois ci un seuil égal à 27 puis 29. Nous avons obtenu les résultats suivants :

- ✓ Seuil =27

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 42,85 % | 81,82 % |
| dessins animés | 50% | 85,71 % |
| Football | 66,67 % | 66,67 % |
| Documentaire | 100 % | 100 % |
| Publicité | 50 % | 66,67 % |
| Journal télévisé | 60 % | 75 % |

Tableau 4.3 : Résultats obtenus en appliquant un seuil égale à 27

✓ Seuil =29

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 50 % | 81,82 % |
| dessins animés | 54,54 % | 85,71 % |
| Football | 66,67 % | 66,67 % |
| Documentaire | 100 % | 100 % |
| Publicité | 50 % | 66,67 % |
| Journal télévisé | 60 % | 75 % |

Tableau 4.4 : Résultats obtenus en appliquant un seuil égale à 29

A partir des résultats du tableau 4.3 et 4.4, nous pouvons remarquer qu'il y a une petite amélioration pour la vidéo *film* et *dessins animés*. Ceci est interprété par la mesure de similarité utilisée entre les arbres quaternaires. Ces valeurs diminuent peu les valeurs du rappel et la précision lorsqu'il y a un faible changement du contenu visuel entre deux images. Nous pouvons déduire que ces valeurs de seuil n'affectent pas la détection de changement de plans brusques et n'améliorent pas la détection de changement de plans progressifs, comme dans le cas de *journal télévisé.*

✓ Seuil =30

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 52,94 % | 81,82 % |
| dessins animés | 60 % | 85,71 % |
| Football | 75 % | 100 % |
| Documentaire | 100 % | 100 % |
| Publicité | 100 % | 100 % |
| Journal télévisé | 60 % | 75 % |

Tableau 4.5 : Résultats obtenus en appliquant un seuil égale à 30

Une amélioration nette des résultats est remarquée pour la majorité des types de vidéos par rapport aux tableaux 4.1, 4.2 et 4.3. Ce qui nous permet de déduire que la valeur du seuil 30 avec la combinaison des trois descripteurs utilisés par notre

algorithme donne des résultats intéressants. Nous avons essayé de vérifier si l'augmentation de la valeur du seuil allait engendrait de meilleurs résultats ? Les résultats obtenus pour des seuils de valeur 31 et 33 respectivement sont présentés dans les tableaux 4.6 et 4.7.

- ✓ Seuil =31

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 50 % | 72,72 % |
| dessins animés | 77,77 % | 100 % |
| Football | 75% | 66,67 % |
| Documentaire | 100 % | 100 % |
| Publicité | 33,34 % | 33,34 % |
| Journal télévisé | 60 % | 75 % |

Tableau 4.6 : Résultats obtenu en appliquant un seuil égale à 31

- ✓ Seuil =33

| Vidéo | Précision | Rappel |
|---|---|---|
| film | 50 % | 63,63 % |
| dessins animés | 87,5 % | 100 % |
| Football | 33,34 % | 33,34 % |
| Documentaire | 100 % | 100 % |
| Publicité | 33,34 % | 33,34 % |
| Journal télévisé | 100 % | 75 % |

Tableau 4.7 : Résultats obtenu en appliquant un seuil égale à 33

D'après les tableaux 4.6 et 4.7, une dégradation des résultats est détectée pour la majorité des types des vidéos. Cela est dû aux valeurs de la mesure de similarité utilisées entres les arbres quaternaire représentant les images des vidéos. Par contre, une amélioration est obtenue pour la vidéo *journal télévisé* qui contient des transitions progressives de type chainé fondu. Les résultats du tableau 4.7 sont traduits par l'approche locale qui correspond à l'arbre quaternaire utilisée par notre technique.

En d'autres termes, ces valeurs permettent d'étendre le taux de changement du contenu visuel entre deux images successives, ce qui permet de détecter les plans progressifs de type fondu enchainés. La figure 20 présente un exemple de la détection des plans de la vidéo documentaire.


Figure 4.3 : Exemple pour la détection des changements de plans par notre algorithme pour la vidéo documentaire

## 4.6 Conclusion

Nous avons présenté dans ce chapitre l'application de notre algorithme de segmentation temporelle de vidéos par arbre quaternaire équilibré à trois niveaux. Les nœuds de chaque niveau (de 0 à 3) stockent un type de descripteur visuel. Ce dernier correspond au contour (obtenu par le filtre de Sobel), la texture (EHD de MPEG-7) et la couleur (SCD de MPEG-7). Les résultats obtenus montrent les performances de notre méthode pour la détection de changement de plans dans le cas des vidéos présentant des changements de plans brusques pour plusieurs valeurs de seuil. Par contre, les résultats sont moins performants dans le cas où les vidéos présentent des transitions progressives de type fondu enchainés. Ceci s'explique par l'utilisation de la structure d'arbre quaternaire et les descripteurs qui sont sauvegardés sur différents niveaux de l'arbre permettant de réaliser l'indexation de l'image. Chaque arbre est associé à une distance. Cette distance influe grandement sur la similarité entre les images du fait qu'elle est calculée à partir de trois types de descripteurs contour, texture et la couleur d'une part ; et d'autre part les caractéristiques propres pour chaque descripteur utilisé jouent un rôle estimé pour un faible changement de plan. L'approche de l'arbre quaternaire permet de faire la recherche locale des images similaires ce qui entraine d'appliquer une certaine exactitude et avec un simple changement inter-quadrant implique une nouvelle transition entre les plans vidéo.

# CONCLUSION GENERALE ET PERSPECTIVES

L'objectif principal de notre travail consiste à utiliser la structure d'arbre en recherche des images par la similitude visuelle dans le but de développer une technique de segmentation temporelle en plans de la vidéo. La technique proposée est basée sur un algorithme qui permet d'avoir les plans de la vidéo à partir d'un ensemble d'images obtenu par l'opération de découpage de la vidéo. L'algorithme exploite l'arbre quaternaire comme structure de données pour représenter les images successives de la vidéo. Cette indexation consiste à sauvegarder les descripteurs visuels à différents niveaux de l'arbre quaternaire. Une distance de similarité est définie afin de regrouper des images successives en des unités élémentaires souvent connues sous le nom de ''plans vidéo''. La procédure de regroupement d'images en plan joue le rôle indispensable et indissociable dans toutes les méthodes de traitement de la vidéo tel que la similitude entre les plans.

La fiabilité des résultats de toute méthode de traitement de la vidéo dépend de la précision de détection des plans. Selon la méthode d'évaluation utilisée '*Rappel et Précision'*, la technique proposée montre des résultats intéressants lorsque le contenu de la vidéo présente des changements de plan brusque. Toutefois, le résultat est moins performant lorsqu'il y a une transition progressive de type chainé fondu. Ceci s'explique par les descripteurs qui sont sauvegardés aux différents niveaux de l'arbre permettant de réaliser l'indexation de l'image et l'association à une distance. Cette distance influe grandement sur la similarité entre les images du fait qu'elle est calculée à partir de trois types de descripteurs contour, texture et la couleur et joue un rôle estimé pour un faible changement de plan.

Une amélioration notable de notre travail serait d'exploiter d'autres descripteurs de MPEG-7 tels que Dominant Color Descriptor (DCD), Homogeneous Texture Descriptor (HTD). En outre, la distance Q-similarité permet de regrouper un

ensemble d'arbres quaternaires ayant des distances similaires selon un seuil. Le choix du seuil influe grandement sur la qualité des résultats ; ce qui nous amène à prévoir l'automatisation de ce seuil.

Nous avons présenté un algorithme de segmentation en plan de la vidéo en utilisant la structure d'arbre quaternaire, cette structure nous permet d'envisager la gestion et le stockage des images de chaque plan vidéo dans la structure de l'arbre quaternaire générique. Comme perspective, cette dernière structure permet de minimiser non seulement l'espace de stockage mais aussi de faire l'indexation et la recherche des vidéos par le contenu visuel.

# REFERENCES BIBLIOGRAPHIQUE


1. K..Haddada , M.Ali Mahjoub « Indexation d'images de cellules sanguine par arbres quartenaires généralisés »,Sciences Of Electronics,Technologies Of Information And Telecommunications Sousse 21-24 mars 2012.

2. R.Hammoud « Construction et présentation des vidéos interactives », Thèse de doctorat. Institut National Polytechnique de Grenoble. 2001.

3. I.Daoudi « Recherche par similarité dans les bases de données multimédia : application
à la recherche par le contenue », thèse de doctorat,2008.

4. A.Guttman, « R-tree : A dynamic index structure for spatial searching », In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 47-57, Boston, MA, June 1984.

5. J.landré « analyse multi résolution pour la recherche et l'indexation d'images par le contenu dans les bases de données images application à la base d'images paléontologique trans'tyfipal », page 68 -69 thèse de doctorat décembre 2005.

6. G.Braviano. « Logique floue en segmentation d'images: seuillage par entropie et structures pyramidales irrégulières », thèse de doctorat à L'Université Joseph Fourrier-Grenoble 1. Octobre 1995.

7. M. H. Yang, et N. Ahuja. « Detecting human faces in color images », Proc. IEEE Int'l Conf. Image Processing, vol 1, pp 127-130, 1998.

8. Park D., Park J., Han J.H.(1999), « Image indexing using color histogram in the CIELUV color space », Proc. of the 5th Japan-Korea Joint Workshop on Computer Vision, pp. 126-132, January 1999.

9. Gong Y., Proietti G., Faloutsos C.(1998), « image indexing and retrieval based on human perceptual color clustering », Proc. Of International Conference on Computer Vision and Pattern Recognition(CVPR), June 1998

10. H.abed,L.Zaoui,Z.Guezzen « Fusion couleur texture dans l'indexation et la recherche des images » JIG'2007 3 émes journées internationales sur l'informatique graphique

11. R.M. Haralick. « statistical and structural approachs to texture ». In proceeding of the IEEE. Pages 786-804, may 1979, number 5, vol.67.

12. B.Burke-Hubbard, « Onde et ondelletes,La Saga d'un outils mathematique ».pour la science 1995.

13. S.Mallat, « A wavelet tour of signal processing Academic », press 1999.

14. Zhang D. and G. Lu. « Review of shape representation and description techniques.Pattern Recognition », 37:1-19, 2004.

15. L.Cieplinski,w-y.Kim, J-R.Ohm,M.Pickering,A.Yamada,«Test of ISO/IEC 15938-3/FDIS information technology-Mulimedea content description interface-part3 visual »,ISO/IEC JTC1/SC29/WG11,MPEG01/N4358, juillet 2001.

16. ISO/IEC JTC1/SC29/WG11N5525 coding of moving pictures and audio « MPEG-7 overview (version 9) », Pattaya ,March 2003.

17. T-B.Zaharia, « Indexation de video et de maillage 3D dans le contexte MPEG-7, these de doctorat, Centre universitaire des Saints-Pères, université Rene descartes - Paris v 2001.

18. I.Daoudi « Recherche par similarité dans les bases de données multimédia : application à la recherche par le contenue », thèse de doctorat,2008.

19. S.Bissol, « Indexation symbolique d'images : une approche basée sur l'apprentissage non supervisé de régularités », thèse de doctorat université joseph fourier –grenoble 1, 2005.

20. S.Jeannim « MPEG-7 visual pary of experimentation model, Version 9.0 »,dans ISO/IEC JTC1/SC29/WG11/N3914,55th Mpeg Meeting. Pisa, Italy.2001.pages 27, 39, 41.

21. W .Y. Kim,Y.S.Kim « A new region-based shape descriptor », TR15-01,December 1999.

22. N.Idrissi, « La navigation dans les bases d'images : prise en compte des attributs de texture », these de doctorat, université de Nantes école polytechnique de l'université de Nantes,2008.

23. N-A.Thacker,F-J.Aherme,P-I.Rocket, « The Bhattacharayya metric as an absolute similarity measure for frequency codec data », 1998 .

24. Y.rubner,C.Tomasi,L-j.Guibas, A metric for distributions with application to images databases proceeding of the 1998, IEEE Internationnal conference of computer vision, Bombay, india.

25. Sid-Ahmed Berrani. « Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision; application à la recherche d'images par le contenu »,2004.

26. M. Flickner, H.Sawhney, J.Ashley, Q.Huang, B.Dom, M. Gorkani,j.Hafner,D.Lee, D.Petkovic, D. Steele et P. Yanker. « query by image and video content: the qbic System ». IEEE computer, 28(9):23–32, 1995.

27.I-R.Bach,C.fuller,A.Gupta,A.Hamparur,B.Harowitz,R.humphery,R.Jain,C-F,Shu,«The virage images earch engine, proceeding of SPIE conference of storage and retrieval for image ardvide database, 1996.

28. A. Pentland, r. W. Picard et s. Sclaroff, « Photobook: Content-based manipulation of images databases ». Technical Report 255, MIT Media Laboratory Perceptual Computing, novembre 1993.

29 R.Sriram,M.Joseph. Francos et A.William, Pearlman. « Texture coding using a wold decomposition model ». In Proceedings of the 12th IAPR International Conference on Pattern Recognition,pages 1382–1386, 1994.

30. Wei-Ying.MA et B. S. Manjunath. Netra: « A toolbox for navigating large image databases », Multimedia Systems, 7(3):184–198, 1999.

31. C.Chad , T.Megan,B.Serge , J. M. Hellerstein et J.Malik, « Blobworld: A system for region-based image indexing and retrieval ». In Proceedings of the 3rd International Conference on Visual Information and Information Systems (VISUAL'99), pages 509–516, London, UK, 1999. Springer-Verlag.

32. N. Boujemaa, J.Fauqueur, M. Ferecatu, F. Fleuret, V.Gouet, B.L.Saux et H.Sahbi, « Ikona for interactive specific and generic image retrieval ». In Proceedings of the International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'01), Rocquencourt,France, 2001.

33. C.Vertan et N. Boujemaa. « Using fuzzy histograms and distances for color image retrieval ». In Proceedings of CIR'2000, Brighton, UK, mai4-5 2000.

34. F.Souvannavong « indexation et recherche de plans vidéo par le contenu sémantique», thèse de doctorat en en traitement de signale et des images, 2005 Télécom Paris.

35. S.Lefevre « détection d'événements dans une séquence vidéo »,these de doctorat en informatique ,2001-2002 université Francois Rabelais Tours page 19.

36. S.Lefevre, « detection d'evenements Dans une sequence video », these de doctorat, Université François Rabelais Tours,2008

37. L.Zaoui ,H.Abed « Stockage et indexation des vidéos par des structure arborescentes » Revue Méditerranéenne des Télécommunications vol. 1, n°2, juillet 2011

38. R.G.Tapu « Segmentation and structuring of video documents for indexing applications», thèse de doctorat conjoint Telecom Sudparis et l'université Pierre et Marie Curie 2012 .

39. Tonomura Y, Abe S. « Content oriented visual interface using video icons for visual database systems ». Journal of Visual Languages and Computing 1990;1(2):183–98.

40. A.Nagasaka, Tanaka Y. « Automatic video indexing and full-video search for object appearances ». In: IFIP Working Conference on Visual Database Systems, Budapest, Hungary, October 1991. p. 113–27.

41. M.Sekma, A. Ben Abdelali Abdellatif Mtibaa, « Application d'un descripteur MPEG7 de texture pour la segmentation temporelle de la vidéo », Sciences Of Electronics,Technologies Of Information And Telecommunications Sousse 21-24 mars 2012 tunisia.

42. M. Manouvrier « Objets similaires de grande taille dans les bases de données», Thèse de Doctorat en Informatique. Université Paris IX-Dauphine. 2000.

43. T.Sellis, N.Roussopoulos et F.Christos, « The R+-tree : a dynamic index for multi-dimensional objects », In proceedings of the 16th International Conference on Very Large databases, page 507-518, 1987.

44. N.Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger, « The R*tree : an efficient and robust access ,method for points and rectangles », In proceeding of the ACM SIGMOD International Conference on Management of Data, page 322-331, 1990.

45. D.A.White, J.Ramesh , « Similarity indexing with the SS-tree », In proceeding of the 12th International Conference on Data Engineering, page 516-523, 1996.

46. N.Katayama, S.Shinichi, « The SR-tree :''An index structure for high dimentional nearest neighbor queries », In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 369-380, Tucson, Arizon USA, 1997.

47. S.Berchtold, D.. A.Keim, Hans-Peter Kriegel, « The X-tree: An index structure for high-dimensional data » , In Proceedings of the 22nd International Conference on Very Large Databases, pages 28–39, 1996.

48. Jon.L.Bentley ,member ,IEEE « Multidimemtional binary search trees In database applications », IEEE Transactions on Software Engineering,Vol.SE-5,NO.4, July 1979.

49. John.T.Robinson. « The k-d-b-tree: A search structure for large multidimensional dynamic indexes ». In Proceedings of the ACM SIGMOD International Conference on Management Data, pages 10–18, 1981.

50. A.Henrich, Hans-Werner Six, P.Widmayer, « The LSD tree: spatial access to multidimensional point and non point objects », In Proceedings of the 15th International Conference on Very large data bases, pages 45-53, Amsterdam, The Netherlands, July 1989.

51. V. Gaede and O. Günther. « Multidimensional access methods ». ACM Computing Surveys, 30(2):170–231,1998.

52. H.Samet. « The design and analysis of spatial data structures ». Addition Wesley, 1989.

53. Samet H., « The Quadtree and related hierarchical structures », Computing Surveys, vol. 16,n 2, 1984, p. 187–260.

54. Manouvrier M., Rukoz M., Jomier G., « Quadtree representations for storage and manipulation of clusters of images », Image and Vision Computing, vol. 20,n 7, 2002, p. 513–527.

55. Luh.,ooib.-C.,Tank.-L., « Efficient image retrieval by color contents », First Int. Conf. on Applications of Database (ADB-94), Vadstena (Sweden), juin 1994, Lecture Notes in Computer Sciences -819 -Springer Verlag.

56. Lin S., Tamer Özsu M., Oria V., NG R., « An extensible hash for multi precision similarity querying of image databases », Proc. of the 27th Int. Conf. on Very Large DataBase (VLDB'2001),Roma (Italy),2001.

57. Marta Rukoz Maude Manouvrier, ,Geneviève Jomier « Distance de similarité d'images basées sur les arbres quaternaires » ,18 èmes journées base de données avancées 21-25 octobre 2002 ,BDA'2002 , page 1-20.

58. Weber, R., Schek, H., and Blott, S. (1998). « A quantitative analysis and performance study for similarity search methods in high-dimensional spaces ». In 24th International Conference on Very Large Databases, New york, NY, USA.

59. Böhm, C., Berchtold, S., and Keim, D. (2001). « Searching in high-dimensional spaces – index structures for improving the performance of multimedia databases ». ACM Computing Survey, 33(3) :322 373.

60. A.Wiktin, « Scale-space filtering » , proc internat joint conference on artificial intelligence, Karlsruhe, Germany , 1983 pp i9-1021.

61. O.R.Vincent, O.Folorunso « A descriptive algorithm for Sobel image edge detection » Proceedings of Informing Science & IT Education Conference (InSITE) 2009.

62. G.Hao Chen1, C.Ling Yang1, L.Man Po2, S.Li Xie « Edge-based structural similarity for image quality assessment », ICASSP. 2006

63. R.Benmokhtar « Fusion multi-niveaux pour l'indexation et la recherche multimédia par le contenu sémantique »,thèse de doctorat, Ecole Doctorale d'Informatique, Télécommunications et Electronique de Paris (EDITE) France. 2009.

64. P.Wu, Y.Man, Ch.Won, Y.Choi, « Texture descriptors in MPEG-7 », 9th International Conference on Computer Analysis of Images and Patterns CAIP'2001. Warsaw, Poland, September 5-7, 2001.

65. M.V.Wembek, « reconnaissance et suivi de visage et implemetation en robotique temps-reel » Mémoire de fin d'études, page 7, Université de Louvain 2009-2010.

66. A.Khalfi,N.Benblidia,A.Cherif-zahar,B.Adidou, « Segmentation en plans de vidéos basée sur la méthode d'arbre et les descripteurs visuels » les 3ème journées doctorales en informatique, organisées par le laboratoire des sciences et technologies de l'information et de la communication, université de Guelma,4,5 décembre 2013.

67. http://www.aoaophoto.com/video-to-picture-converter/video-to-picture.htm

68. M.Guironnet « Méthodes de résumé de vidéo a partir d'informations bas niveau, du mouvement de camera ou de l'attention visuelle ». Thèse de doctorat. Université joseph Fourier – grenoble 1 France. 2006.